Project no. FP6-027178

VISP

Virtual ISP

Instrument: STREP
Thematic Priority: IST (FP6)

D2.2 – Workflow Software Analysis and Comparison

Due date of deliverable: 28 February 2006
Actual submission date: 27 March 2006

Start date of project: 1st November 2005. Duration: 32 months

Editor: Todor Stoilov (ICCS)

Contributors:

Benoit Gaillard (Valtech), Boriana Vachova (ICCS), Dana Petcu (WUT), Daniel Pop (WUT), Elena Ivanova (ICCS), Elisaveta Trichkova (ICCS), Florin Fortis (WUT), Francesco Fornasari (Metaware), Klaus-Peter Eckert (FhI Fokus), Krasimir Trichkov (ICCS), Krasimira Stoilova (ICCS), Krzysztof Samp (UAM), Martin Tsenov (ICCS), Moritz Rüsch (FhI Fokus), Rafal Renk (UAM), Simeon Cvetanov (ICCS), Todor Stoilov (ICCS), Yuri Glickmann (FhI Fokus)

Revision: 1.0

Project co-funded by the European Commission within the Sixth Framework Programme (2002-2006)		
Dissemination Level		
PU	Public	**X**
PP	Restricted to other programme participants (including the Commission Services)	
RE	Restricted to a group specified by the consortium (including the Commission Services)	
CO	Confidential, only for members of the Consortium (including the Commission Services)	

Audience

This document is public and is intended to the following audience:

- VISP partners
- WP03, WP10 members to carry on next steps

Document history

This document will be kept up to date during the duration of the project.

Version	Date	Author	Partner	Comments
1.0	27-03-06	Todor Stoilov	ICCS	First version

Executive summary

This deliverable addresses the set of problems, related to the assessment and application of software products, which are used for the design, and exploitation of workflow management systems. The deliverable draws the attention towards the available experience of assessing and comparing such software suits. The works in deliverable D2.2 have been conducted in the following sequence.

First it has been identified the importance of the automation for the design, implementation and exploitation of workflow management systems. Next, it has been performed surveys, which present the current state about workflow targets in automation, representatives of workflow tools, their classifications, available results in the product evaluation, and current practice in software evaluation. The surveys give ground for the project development, which are presented in Chapters 5 and later. Using the overview descriptions, it has been defined the functionalities, which have to be supported by the workflow software products. Appropriate classifications of the products have been performed, concerning their internal software origin, their sets of functionalities and targeted domains of implementation. It has been identified the problems, related to the comparison and evaluation of different software products and the experience, available for the assessment of relatively different software suits. The lack of direct quantitative evaluations of the products insisted the works in this deliverable to be directed towards getting personal experience of them. Thus, the deliverable works involve installation, configuration and trial test of a several software products, carefully chosen among a set of candidate technologies. By testing the products, we get personal experience for their potential, which benefits the assessment and comparison of the products. The problem, which arises, was to minimize the subjective influence of the testing experts in their personal evaluation findings. An idea to overcome this problem, developed in this deliverable, was to apply a common evaluation scheme, which is based on objective requirements towards the products. We found such common basis by using the standard ISO/IEC 9126 for the quality of the software products. Following the general requirements of this standard a template has been designed, which gives a common framework for the evaluation of the software products. The results from the assessment have been summarized and appropriate classification is performed, leveraging the decisions about the prospective set of products, which have to be used for the VISP developments.

This project has received research funding from the European Community's Sixth Framework Program. This document reflects only the author's views. The European Community is not liable for any use that may be made of the information contained therein.

Abbreviations

API	Application Program Interface
BAM	Business activity monitoring
ADSL	Asymmetric Digital Subscriber Line
ASAP	Asynchronous Application Service Protocol
AWS	Active Webflow Standard
B2B	Business-to-Business
BPDM	Business Process Definition Metamodel
BPEL	Business Process Execution Language
BPEL4WS	Business Process Execution Language for Web Services
BPM	Business Process Management
BPMI	Business Process Management Initiative
BPML	Business Process Modelling Language
BPMN	Business Process Modelling Notation
BPMS	Business Process Management Systems
BPQL	Business Process Query Language
BPSS	Business Process Specification Schema
BSF	Bean Scripting Framework
CORBA	Common Object Request Broker Architecture
CPA	Collaboration Protocol Agreement
CPP	Collaboration Protocol Profile
CRM	Customer Relationship Management
DB	DataBase
DBMS	Database Management Systems
DPQ	Delivered Product Quality
DQ	Design Quality
EAI	Entreprise Application Integration
ebXML	Electronic Business using eXtensible Markup Language
ECM	Enterprise Content Management
EJB	Entreprise Java Bean
EPC	Event-driven Process Chains
EPQ	Estimated Product Quality
ERP	Enterprise Resource Planning
ESB	Enterprise Service Bus
FDL	Flow Definition Language
FTP	File Transfer Protocol
GNU	GNU's Not UNIX
GPL	General Public License
GQ	Goal Quality
GUI	Graphical User Interface
HTML	HyperText Markup Language
HTTP	Hypertext Transfer Protocol
IMAP	Internet Message Access Protocol
IT	Information Technology
Java API	Java Application Program Interface
JBPEL	JavaBusiness Process Execution Language
jBPM	Java Business Process Management

JCA	Java Connector Architecture
JCP	Java Community Process
J2EE	Java 2 Platform, Enterprise Edition
JDBC	Java DataBase Connector
JMS	Java Message Service
JMX	Java Management Extensions
JNDI	Java Naming and Directory Interface
J2SE	Java 2 Platform, Standard Edition
jPDL	jBPM Process Definition Language
JSP	Java Server Page
JSR	Java Specification Request
JTA	Java Transaction API
KPIs	Key Performance Indicators
LDAP	Lightweight Directory Access Protocol
LGPL	Lesser General Public License
MDA	Model Driven Architecture
MIB	Management Information Base
MSIE	Microsoft Internet Explorer
MSMQ	Microsoft Message Queuing
OMG	Object Management Group
OS	Operation System
PDD	Process Deployment Descriptor
PDEF	Partner Definitions
PDF	Portable Document Format
PHP	PHP Hypertext Preprocessor
POJO	Plain Old Java Object
POOL	Process Oriented OpenWFE Lisp/Language
POP3	Post Office Protocol
QIU	Quality in Use
QVT	Query View Transformation
RDBMS	Relational Database Management Systems
REST	Representational State Transfer
RMI	Remote Method Invocation
RPQ	Required Product Quality
SAML	Security Assertions Mark-up Language
SCUFL	Simple Conceptual Unified Flow Language
SNMP	Simple Network Management Protocol
SMTP	Simple Mail Transfer Protocol
SOA	Service Oriented Architecture
SOAP	Simple Object Access Protocol
SQL	Structured Query Language
TCP	Transmission Control Protocol
UBL	Universal Business Language
UDA	User-defined attribute
UDAs	User-defined process attributes
UDDI	Universal Description, Discovery and Integration
UML	Unified Modelling Language
URI	Uniform Resource Identifier
URN	Uniform Resource Name
VIS	Versata Integration Server

WAPI	World Association of Professional Investigators
WFM	Workflow Management
WfMC	Workflow Management Coalition
WFMS	Work Flow Management Systems
Wf-XML	Workflow XML
WSCI	Web Service Choreography Interface
WS-CDL	Web Services Choreography Description Language
WSDL	Web Services Description Language
WSFL	Web Services Flow Language
WSIF	Web Services Invocation Framework
WSIL	Web Services Inspection Language
WS-RM	WS-ReliableMessaging
XLANG	XML LANGuage
XMI	XML Metadata Interchange
XML	Extensible Markup Language
XPDL	XML Processing Description Language
XSD	XML Schema Definition
XSLT	EXtensible Stylesheet Language Transformations
YAWL	Yet Another Workflow Language

Table of Content

1. Introduction....12
2. Importance of Workflow Automation....14
2.1. Introduction....14
2.2. Research Areas Relevant to Software Workflow Modelling....14
2.3. The Evolution of Business Process Management....16
2.4. System's and Technological Architecture of Workflow Management Systems....20
2.5. Classes of Workflow Applications....21
2.6. Methodology for the Evaluation of Software Tools....25
3. Suites for Workflow Software....27
3.1. Classifications of Software Tools....27
3.1.1. Classifications of Software Tools for Business and Scientific Workflows....27
3.1.2. Classifications Of Software Tools According To Their Software Language Design....29
3.1.3. Classifications Of Software Tools According To Their Supported Standard....30
3.1.4. Open Source and Commercial Workflow Tools....31
3.2. Current Experience in Evaluation of Workflow Software Products....32
3.2.1. Comparisons And Assessments Of Workflow Software Products And Tools....32
3.2.2. Evaluation of Workflow Software Products According To The Workflow Patterns Supported....39
3. 3 Software Evaluation Methodology....47
3.3.1. Smith's criteria for evaluation of Internet based software and information resources....48
3.3.2. Belyk's And Feist's Software Evaluation Criteria....50
3.3.3. Five Criteria For Evaluating Web Pages....52
3.3.4. WWW CyberGuide Ratings....53
3.3.5. General Software Evaluation Criteria....55
3.4. Conclusions....60
4. Survey of Software Products, Supporting Workflow Management....61
4.1. ActiveBPEL Engine....66
4.2. ActiveWebflow™ Standard....67
4.3. ActiveWebflow™ Professional Designer....71
4.4. Biztalk Server....74
4.5. Borland Together Designer 2006....77
4.6. Cape Clear 6.5....79
4.7. Con:Cern....82
4.8. Enhydra JaWE....84
4.9. Enhydra Shark - Java Open Source XPDL Workflow....88
4.10. EnterpriseXtention....91
4.11. Freefluo....93
4.12. JBoss jBPM....96
4.13. ObjectWeb BONITA....99
4.14. OpenWFE....102
4.15. Oracle BPEL Process Manager....103
4.16 ITP-Commerce Process Modeler for Microsoft Visio....107
4.17 SAP NetWeaver Exchange Infrastructure....109
4.18 WfMOpen....110
4.19 YAWL....113

4.20. Survey of UML Modelling Tools 117
4.21. Assessed Software and Corresponding Covered Technologies 118
5. Software Evaluation Methodologies....119
5.1. ISO/IEC 9126 Evaluation Plan 119
5.2 VISP Methodology for Comparison of the Workflow Software Products 130
5.3. VISP Template For Evaluation Workflow Software Products 134
5.4. Conclusions 146
6. Evaluation Findings 147
6.1 Introduction 147
6.2. Functional Evaluation Of Characteristics Of Short-List Products 147
6.3 Evaluations 149
ActiveBPEL Engine 149
Active Endpoints ActiveWebflow Standard 153
ActiveWebflow Designer 157
BizTalk Server 161
Borland Together Designer 2006....164
Cape Clear 167
Con-cern....171
Enhydra JaWE....174
Enhydra Shark....179
EnterpriseXtention....182
Freefluo 186
JBoss jBPM 188
ObjectWeb BONITA 193
OpenWFE....199
Oracle BPEL Process Manager 201
Process Modeler....204
SAP NetWeaver Exchange Infrastructure 206
WfMOpen 208
YAWL....211
6.4 Comparisons....215
6.5 Conclusions 228
7. CONCLUSIONS 232
REFERENCES....234

Table of Figures

Figure 2.1 - History of Office Automation and Workflow Systems …… 18
Figure 2.2 - Software architecture of a Workflow Application ……. 21
Figure 2.3 - Collaborative workflow ……. 22
Figure 2.4- Production workflow …….. 24
Figure 2.5- Administrative workflow …… 25
Figure3.1 - Workflow Pattern 1 (Sequence) in i-Flow ……. 37
Figure3.2 - Workflow Pattern 1 (Sequence) in MQ Workflow …… 37
Figure3.3 - Workflow Pattern 2 (Parallel Split) in i-Flow …….. 37
Figure3.4 - Workflow Pattern 2 (Parallel Split) in MQ Workflow …….. 38
Figure3.5 - Workflow Pattern 3 (Synchronization) in i-Flow …….. 38
Figure3.6 - Workflow Pattern 3 (Synchronization) in MQ Workflow ……. 38
Figure3.7 - Workflow Pattern 4 (Exclusive choice) in i-Flow ……. 39
Figure3.8 - Workflow Pattern 4 (Exclusive choice) in MQ Workflow …… 39
Figure3.9 - Design patterns for the multi-choice …… 40
Figure 3.10 - Sample process ……... 43
Figure3.11 - Sample process model implemented with Visual WorkFlo …….. 43
Figure3.12 - Sample process model implemented with Forté Conductor ……. 44
Figure3.13 - Sample process model implemented with Changengine …….. 44
Figure3.14 - Sample process model implemented with Staffware …… 45
Figure3.15 - Sample process model implemented with i-Flow …… 45
Figure3.16 - Sample process model implemented with MQSeries Workflow ……. 46
Figure3.17 - Sample process model implemented with Verve ……. 46
Figure 3.18 - Sample process model implemented with SAP R/3 Workflow ……... 47
Figure 4.1 - ActiveBPEL engine …… 66
Figure 4.2 - ActiveWebflow BPEL process …….. 72
Figure 4. 3 - Generated BPEL code …….. 72
Figure 4.4 - Creating processes ……. 74
Figure4.5. Biztalk Server's supported standards …… 75
Figure4.6. Components of the BizTalk Server engine …… 76
Figure 4.7 - Cape Clear 6.5's Structure …… 79
Figure 4.8 - Cape Clear 6.5' Interfaces ……. 82
Figure 4.9 - Concept of the Process Definition Interchange ……. 85
Figure 4.10 - Package level …….. 86

Figure 4.11 - Process level …… 87
Figure 4.12 - Architecture of Enhydra Shark …….. 89
Figure 4.13 - Enhydra Shark process monitor …….. 90
Figure 4.14 - Supported interfaces ……... 95
Figure 4.15 - Taverna workbench …… 95
Figure 4.16 - Example of graph from the editor …….. 97
Figure 4.17 - jBPM database overview …… 98

Figure 4.18 - Bonita GraphEditor tool ……. 100
Figure 4.19 - The Bonita Architecture ……. 101
Figure 4.20 - Components of the Oracle BPEL Process Manager …….. 104
Figure 4.21. SAP NetWeaver Exchange structure ……. 110
Figure 4.22. SAP NetWeaver Exchange structure ……. 110
Figure 4.23 – The workflow engine's architecture ……. 111

Figure 4.24 – The workflow engine's application …….. 113
Figure 4.25 YAWL Specifications …… 114
Figure 4.26 YAWL Engine …… 115
Figure 4.27 - YAWL Editor …… 116
Figure 5.1 - Quality in the life-cycle …….. 121
Figure 5.2 - Software product quality …… 123
Figure 5.3 - Relationship between different types of quality …… 123
Figure 6.1 Pie-charts for Active BPEL Engine's characteristics …….. … 153
Figure 6.2 Pie-charts for ActiveWebflow Standard's characteristics ……. … 157
Figure 6.3 Pie-charts for ActiveWebflow Designer's characteristics ……..… 161
Figure 6.4 Pie-charts for Biztalk Server's characteristics ……..... 164
Figure 6.5 Pie-charts for Borland Together Designer's characteristics ……..… 166
Figure 6.6 Pie-charts for Cape …… 171
Figure 6.7 Pie-charts for Con …….. … 174
Figure 6.8 Pie-charts for Enhydra JaWE's characteristics ……..… 179
Figure 6.9 Pie-charts for Enhydra ……. … 182
Figure 6.10 Pie-charts for …….. 185
Figure 6.11 Pie-charts for Freefluo's characteristics …… 188
Figure 6.12 Pie-charts for …….. 193
Figure 6.13 Pie-charts for …….. 198
Figure 6.14 Pie-charts for ……. 201
Figure 6.15 Pie-charts for Oracle BPEL Process …… 204
Figure 6.16 Pie-charts for Process Modeler for Microsoft …… 206
Figure 6.17 Pie-charts for SAP NetWeaver Exchange Infrastructure's characteristics …….. 208
Figure 6.18 Pie-charts for WfMOpen's characteristics …….. 211
Figure 6.19 Pie-charts for YAWL's characteristics ……. 215
Figure 6.20 Workflow Editors comparative chart …… 217
Figure 6.21 Workflow Engines comparative chart …….. 217
Figure 6.22 Workflow Editors + Engines comparative chart …….. 218
Figure 6.23 Workflow Editors Usability comparative chart …… 218

Figure 6.24 Workflow Engines Usability comparative chart …….. 219
Figure 6.25 Workflow Editors + Engines Usability comparative chart …….. 219
Figure 6.26 Reliability comparative chart …… 220
Figure 6.27 Efficiency comparative chart …… 220
Figure 6.28 Maintainability comparative chart …… 221
Figure 6.29 Portability comparative chart …… 221

List of Tables

Table 3.1: List of nodes and their perspective support for Java Actions ……... 35
Table 3.2 Product Pattern 1 ……. 41

Table 3.3 Product Pattern 2 ……. 42
Table 3.4 Matrix wighted evaluation of the software product …….. 51
Table 3.5 Five criteria for evaluating Web pages ……. 52
Table 4. 1 - Long list of products, supporting standardization initiatives ……. 61
Table 4. 2 - Workflow products, allocated per partners for test and assessment …….. 65
Table 4. 3 - Alphabetic order of the short list products, evaluated by the VISP consortium 65
Table 4. 4 - Functional peculiarities of AWS tool ……. 68
Table 4.5 - Settings of AWS engine ……. 69
Table 4. 6 - Functionalities of ActiveWebflow Designer …….. 72
Table 4.7 – The assessed short-list workflow software and corresponding workflow technologies ……. 118
Table 6.1. General workflow functionality decomposition of the products……. 216
Table 6.2. Average evaluation results per products ……. 222
Table 6.3. Average evaluation results – Modelling Tools……. 222
Table 6.4. Average evaluation – Workflow editors…….. 223
Table 6.5. Average evaluation results – Workflow Engines……. 223
Table 6.6. Average evaluation – Combined Products (Editor+Engines)……. 224
Table 6.7. Modelling Tools …… 229

Table **6.8.** Workflow Editors ……. 229
Table 6.9. Workflow Engines …… 230

Table 6.10. Combined Products: Editor + Engine ……. 231

1. Introduction

Workflow Management Systems are a mature technology for automating and controlling business processes [Kappel , 2000] , [Leymann, 2000]. One widely accepted definition of workflow comes from the Workflow Management Coalition [WfMC, Glossary, 1999]: "Workflow is the computerized facilitation or automation of a business process, in whole or part". An extended definition in [WfMC, Glossary, 1999] works out the definition as *"…a system that defines, creates and manages the execution of workflows through the use of software, running on one or more workflow engines, which is able to interpret the process definition, interact with workflow participants and, where required, invoke the use of IT tools and applications."*

With the rise of the Web as the major platform for making data and services available for both, humans and applications, a new challenge has become prevalent requiring not only the support of workflows within individual organizations (called *intraorganizational workflows*), but also workflows crossing organizational boundaries referred to as interorganizational workflows, [Aalst, 1999].

Although interorganizational workflows are still very much open to research, one can identify two major characteristics which distinguish them from intraorganizational workflows and at the same time lead to several new functional requirements on specification languages, not present in the intraorganizational case. First, *interoperability* is a prerequisite for interorganizational workflows. Interoperability requires agreements on the interfaces between organizations, which provide a common understanding of the data and services exchanged. Interface standardization or interface bridging become necessary in spite of potential heterogeneity of autonomous organizations' interfaces [Wegner, 1996]. Second, the *autonomy* of organizations participating in an interorganizational workflow has to be considered, whereby different kinds of autonomy are relevant at different stages in the lifecycle of the workflow. These comprise design autonomy at build time, communication and execution.

A general task of the development of the workflow system in the current business activities is the implementation of principles of the automatic control for business systems. These systems do not consist of pure technical components, but they integrate both human and human-computer activities and non-automatic interactions. Thus, the implementation of the principles of the automation will benefit the exploitation behaviour of the business systems. The importance of Workflow Automation is presented in **Chapter 2**. In **Chapter 3** are presented suits for Workflow Automation. A kind of evaluations of workflow software products based on workflow pattern is described in Chapter 3. Here are presented sets of evaluation criteria used for assessment of software products, which unfortunately do not cover the versatile characteristics of the workflow software but they are found as a state-of-the-art cases, available now. As the list of the available software includes more than 134 products, which have a different stage of maturity and application in **Chapter 4** it is described the architecture, interfaces, supported standards of the 19 most developed now and appropriate for VISP project Workflow software products. These products are included in a so-called short-list and tested by the consortium members. As unfortunately, the evaluation criteria used for the software products, presented in Chapter 3, do not cover the versatile characteristics of the workflow software, in **Chapter 5** it is proposed another methodology

for evaluation, which applies the standard ISO/IEC 9126 for the quality of software products. A template for the evaluation of the workflow software products by the VISP partners based on ISO/IEC 9126 requirements is developed here. The results from the functional evaluation of the characteristics of the short-list products are presented in **Chapter 6**. It is described the methodology for the comparison of the products proposed and developed by the VISP partners. In the **Conclusion** paragraph, a suggestion is made that for the next project developments the VISP partners are not restricted to use only one product but to have the opportunity to choose more than one workflow software product. These conclusion origins from the fact that it exists now several useful workflow software products appropriate for VISP utilization. These products can combine functionalities in modelling, editing and executing workflow objects and tasks. However, the important issues are that these products have to support common workflow standards, which are used for the development and implementation of workflow systems. The deliverable suggests a set of recommended tools, which are prospective for the VISP project developments.

2. Importance of Workflow Automation

2.1. Introduction

To implement automation techniques and control methods in the business processes it is necessary to apply modelling techniques for the non-technical and organizational systems and to extend the functionalities of the informational computer driven systems in the organizations. Over the last decade there has been increasing interest in information systems that are used to support, control, and/or monitor business processes. Examples of systems that are managed by implicit or explicit process models are Enterprise Resource Planning (ERP) systems, Work Flow Management Systems (WFMS) and Customer Relationship Management (CRM) systems. These systems are implemented to support specific business processes. A set of formal languages have been worked out to support process-orientation in the context of web services (cf. BPEL4WS, BPML, WSCI, etc.). The support of leading firms as IBM, Microsoft, HP and SAP for a language like BPEL4WS (Business Process Execution Language for Web Services), [Andrews, 2003] [Andrews, 2003] Andrews T., F. Curbera, H. Dholakia, Y. Goland, J. Klein, F. Leymann, K. Liu, D. Roller, D. Smith, S. Thatte, I. Trickovic, and S. Weerawarana, Business Process Execution Language for Web Services Version 1.1, 2003. Technical report, Accessed at http://xml.coverpages.org/ BPELv11-May052003Final.pdf reinforces the fact that process-awareness has become one of the cornerstones of information systems development. Existing languages and tools focus on control-flow and combine this focus with mature support for data in the form of XML and database technology. As a result, control-flow and data-flow are well addressed in languages and systems: BPEL4WS, XPDL [Womb, 2002][WfMC, 2002] WfMC, Workflow Process Definition Interface –XML Process Definition Language. Technical Report Document Number WFMC-TC-1025, Workflow Management Coalition, 2002, http://www.wfmc.org/standards/docs.htm, Wixom.

2.2. Research Areas Relevant to Software Workflow Modelling

The “workflow management system” becomes important term. *Workflow* can be defined as "the sequence or steps used in business processes" [Marshak, 1994]. The workflow process has to prescribe that more than one person is involved with their activities and that the workflow consists of both sequential and parallel steps. *Workflow Management* is supporting and controlling the workflow. An important objective in Workflow Management is to automatically route artefacts (documents, messages, e-mails) through a network to actors having predefined roles. The routing is done according to a set of predefined rules, and is often controlled by the state of the artefact (e.g., the price). Business rules are particularly important for routing of artefacts. This approach requires relatively stable business rules, as the rules cannot always be changed on the fly. The business domain of Workflow Management is addressing the information services. The intent of using workflow technology within the Workflow Management presumes to be developed the full range of problems from understanding to automation.

Workflow management deals with supporting business processes in organizations and it involves managing the flows of work through an organization [Aalst, 1998]. Workflows are a collection of coordinated tasks designed to carry out a well-defined complex process [Mukherjee, *et al*., 2004]. A workflow management system is a generic information system

that supports modelling, execution, management and monitoring of workflow. Many different workflow management systems have been developed that focus on different application domains and provide different functionality [Aalst *et al*., 2003]. Workflow management lacks a standardized theory that provides a theoretical background for workflows like the relational algebra provide for databases [Aalst, 1998]. Despite efforts of standardization bodies, there is no consensus on the representation or conceptual model of workflow processes [Sheth, 1999]. In this diverse and complicated landscape, it is a challenge to evaluate and compare the functionality of various workflow management systems and to survey the functional requirements of workflow management systems. A number of approaches attempt to address this situation. In [Jablonski *et al*., 1996] [Jablonski *et al*, 1996] Jablonski S. and C. Bussler, *Workflow Management: Modelling Concepts, Architecture and Implementation,* 1996. International Thomson Computer Press, London. there are described a lot of essential perspectives and aspects of comprehensive workflow management functionality. They provide a structure in the complex environment of workflows. In [Aalst *et al*., 2003] it is systematically analyzed available functionality in existing workflow management systems, and they are categorized in a number of *workflow patterns*. For the evaluation of the software tools, deploying workflow technologies, an integrated view is applied in this deliverable combining both qualitative and quantitative indicators to assess them for their support in workflow management systems.

Many organizations came to the conclusion that although they have adapted the Information Technology (IT) for the improvement of their working efficiency, the business processes within their organizations and between themselves and their partners have not been clearly described and streamlined [Yan et al, 2001]. During the execution of business processes, there are not enough techniques and methods to monitor and control the processes. This leads to the misunderstanding of responsibilities, blocks in coordination, and slow reaction to the changing market. Workflow management technology is among the ones under development to overcome these shortcomings. Workflow management comes from office automation area, where all kinds of documents need to be digitized and transferred among co-workers. Nowadays, the workflow management attracts a lot of attention due to its ability in modelling, executing, and monitoring processes. The processes can be not only the business processes, but also any procedures that need to be presented and managed. But the motivation for its booming is its promising usage in managing business processes. The benefits of applying workflow technology to business process management are as follows [Yan *et al*., 2001]:

- The business processes are explicitly defined, so that the responsibilities and the coordination relations are clearly determined.
- It is easy to optimize the business processes because of their explicit definitions.
- Business processes are modularized and these modules can be reorganized by the WFMS to form new business processes, so as to react quickly to unexpected changing business needs and conditions.
- WFMS can track daily operations.
- WFMS integrates applications on different platforms into a business process.
- WFMS provides personal workplaces.
- WFMS separates business logic of a process from the tasks themselves in the process.

Therefore, the user in the workflow system does not have to deal with the route of the business process, but only concentrates on the task itself. From a technical perspective, WFMS bring together principles, methodologies and technologies from various areas of computer science, control theory, automation and management science. For example,

workflow techniques involve database management, client-server computing, heterogeneous distributed computing, graphical user interfaces, application and subsystem integration, messaging, document management, simulation, and business practices and re-engineering. The actuality of the WFMS is approved by the attempts of developing such systems for e-business services. A new funded EC project [DBE06] tries to develop platform on which other projects can develop e-Business solutions. Thus, the DBE project contributes to the domain of integration of distributed e-business solutions. This is targeted also by VISP project. However, in VISP the integration is achieved through the design of a common workflow business system.

2.3. The Evolution of Business Process Management

One of the leaders in development of workflow technology is Wil van der Aalst [Aalst, 1998; Aalst, 1999; Aalst *et al*. 2000; Aalst and Hee, 2002; Aalst *et al*., 2003]. In [Aalst *et al*, 2003] he makes a history review of the workflow technology. There are examples of workflow management systems such as Staffware, enterprise resource planning systems, such as SAP and Baan, and other specific systems. Only separate consultants and companies are interested in business processes. There is missing a more fundamental approach until the nineties, considered now as the beginning of researchers' work on the foundations of business process' management systems.

"The aim of workflow management technology is the separation of process logic from application logic, in order to enable flexible and highly configurable applications" [zur Muehlen, 2004]. In [Jablonski, *et al*, 1996] it has been defined seven fields as the *conceptual ancestors* of workflow management technology: office automation, database management, e-mail, document management, software process management, business process modelling, and enterprise modelling and architecture. The focus of office automation research was "to reduce the complexity of the user's interface to the (office information) system, control the flow of information, and enhance the overall efficiency of the office" [Ellis and Nutt, 1980]. An overview of the historical development of office automation systems and workflow technology is given in figure 2.1 [zur Muehlen, 2004]]. "While the research interest in office automation increases since middle of the 1980, the commercial exploitation of workflow technology began between 1983 and 1985. It was fostered by advances in imaging and document management technology on the one side, and enhanced e-mail systems that extended traditional point-to-point mail routing with a predefined process map on the other side. From this first generation of workflow systems, only few vendors are still active, while the majority of the early players have been restructured through mergers and acquisitions, or dropped out of the market altogether" [zur Muehlen, 2004]. The fundamental understanding of workflow management and workflow management systems has been determined mainly by the terminology and glossary work of the Workflow Management Coalition [WfMC, Glossary, 1999]. According to it the workflow management system consists of a modelling component for the creation of workflow models, functionality for the creation of workflow instances from these workflow models, and functionality for the execution of these workflow instances. Respectively the products, which implement the workflow technology, have to follow this functional consistence with appropriate software modules. The functional component for the enactment of workflow instances is called *workflow engine.*

In the sixties information systems were built on top of a small operating system with limited functionality. These systems mainly consisted of tailor-made applications. New types of

product software added new functionalities since then. The nowadays-operating systems offer much more functionality than in the sixties. Database management systems offer functionality, which used to be in tailor-made applications. This trend leads to the shifted emphasis from programming to assembling of complex software systems. The coding of individual modules is already an old approach. The challenge now is orchestrating and combining pieces of software from the different functional layers. Another trend is the shift from data to processes [Aalst *et al*., 2003]. The same source points out the seventies and eighties as dominated by data-driven approaches. At that period, the information technology is associated with storing and retrieving information and as a result, data modelling was the starting point for building an information system. The processes had to adapt to information technology and modelling of business processes were often neglected. On the contrary, today the emphasis is on processes, for instance business process reengineering.

Another development trend is the shift from carefully planned designs to redesign and organizational growth [Aalst *et al*., 2003]:
"*Business process management systems* are information systems that are "process aware", i.e., systems that go beyond the automation of individual tasks. Notable examples of business process management' systems are workflow management systems such as Staffware, MQSeries, and COSA, and case handling systems such as FLOWer. Note that leading enterprise resource planning systems also offer a workflow management module. The workflow engines of SAP, Baan, PeopleSoft, Oracle, and JD Edwards can be considered as integrated business process management' systems. The idea to isolate the management of business processes in a separate component is consistent with the three trends identified. Business process management' systems can be used to avoid hard coding the work processes into tailor-made applications and thus support the shift from programming to assembling. Moreover, process orientation, redesign, and organic growth are supported. For example, today's workflow management systems can be used to integrate existing applications and support process change by merely changing the workflow diagram.

Workflow Management Systems Workflow management systems use a large variety of languages and concepts based on different paradigms. Most of the products available use a proprietary language rather than a tool-independent language. Some workflow management systems are based on Petri nets but typically add both product specific extensions and restrictions. Other systems use a completely different mechanism. For example, IBM's MQSeries workflow uses both active and passive threads rather than token passing. The differences between the various tools are striking. One of the reasons attributed to the lack of consensus of what constitutes a workflow specification is the variety of ways in which business processes are otherwise described. The absence of a universal organizational "theory", and standard business process modelling concepts, it is contended, explains and ultimately justifies the diversity in workflow languages and the major differences between them. What is more, the comparison of different workflow products winds up being more of a dissemination of products and less of a critique of workflow language capabilities.

Web Services Composition Languages There are two trends coming together in the world of E-business that are creating both opportunities and pressures to automate business processes across organizational boundaries. One is the technology push created by enabling technologies taking XML-based standards and the Internet as a starting point. The other trend is the need to improve the efficiency of processes from a business perspective. After the dotcom crash, there is a pressing need to truly utilize the potential of Internet technology by automating business processes across enterprise boundaries. The goal of web services is to

exploit XML technology and the Internet to integrate applications than can be published, located, and invoked over the Web. A typical example of a web services application is the Galileo system that connects more that 42,000 travel agency locations to 37 car rental companies, 47,000 hotels, and 350 tour operators. To truly integrate business processes across enterprise boundaries, it is not sufficient to merely support simple interaction using standard messages and protocols. Business interactions require long-running interactions that are driven by an explicit process model. This raises the need for web services composition

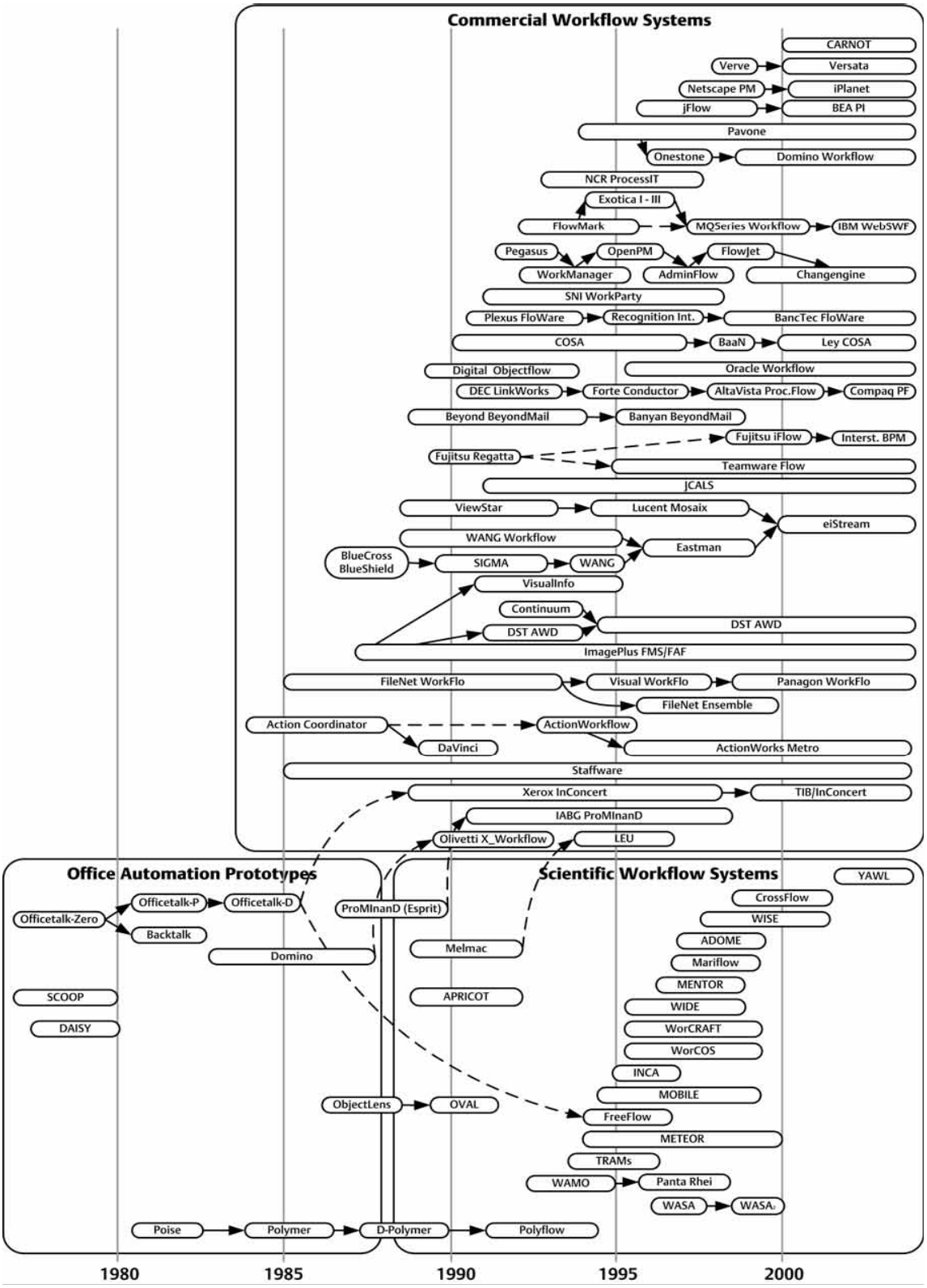


Figure 2.1 - History of Office Automation and Workflow Systems[1]

[1]Copied from [zur Muehlen, 2004]

languages such as BPEL4WS, WSFL, XLANG, WSCI, and BPML.

Development with respect to web services composition languages have been mainly driven by software vendors like IBM, Microsoft, Sun, BEA, SAP, and Intalio. This has resulted in an abundance of standards having overlapping functionality. When looking at the standards in more detail, it is clear these are often based on existing products. A good example is WSFL, which is almost a copy of IBM's Flowmark/MQ Series Workflow language. Standards, which involve multiple software vendors, are often a compromise between competing viewpoints. As a result, such standards tend to be imprecise or unnecessarily complex. WfMC's XPDL is an example of a standard, which is an imprecise thereby allowing vendor to have his or her own interpretation of the standard (thus making the standard useless). BPEL4WS joins viewpoints from both WSFL and XLANG thus making the language very complex. Given these observations it is useful to look for objective measures for comparing web services composition languages because this comparison will give ground for the evaluation and assessment of corresponding software tools, which implement the different modelling and evectional standards in the workflow domain".

One of the conclusions in [Aalst *et al*., 2003] is that the world is confronted with too many standards, which are mainly driven, by concrete products and/or commercial interests. The efforts of specialists are directed to narrow the launch of software workflow tools by ignoring standardization proposals that are not using well-established process modelling techniques.

The first standardization effort in workflow was the Workflow Management Coalition (WfMC) [2.1]. In its reference model [2.2] it was defined the interfaces between a workflow management system and other actors. Another development of WfMC is XPDL specification [2.3], defining an XML schema for specifying the declarative part of a workflow. Another initiative that standardizes the automation of business processes on a J2EE server is JSR 207 (Java Specification Request) [2.4]. In June of 2005, the Business Process Management Initiative (BPMI.org) and the Object Management Group™ (OMG™) announced the merger of their Business Process Management (BPM) activities to provide thought leadership and industry standards for this vital and growing industry. The combined group has named itself the **Business Modelling & Integration (BMI) Domain Task Force (DTF)**. The BMI's combined activities continue BPMI's and OMG's groundbreaking work and focus on all aspects of Business Process Management. BPMI's widely used standard for business modelling, **Business Process Modelling Notation (BPMN)**, and the **Business Motivation Model (BMM)** contributed by the Business Rules Group, have completed all but the final vote of the OMG adoption process [2.5]. A joint initiative of OASIS and UN/CEFACT is ebXML's BPSS (Business Process Specification Schema) - standards for business collaboration [2.6]. BPMI's BPML & WSCI defines a specification (BPMN), which describe how the "executional" business processes can be visually represented [2.7]. OMG's Workflow management facility is based upon the WfMC specifications and defines how this gets translated into CORBA [2.10]. Different standards have been developed like UML [2.11] for modelling and designing software systems; RosettaNet [2.12] - for Partner Interface Processes; UBL the Universal Business Language (UBL) [2.13] – a standard library of XML documents for communication between different organizations.

The standardization efforts try to extend the application area of e-Business solutions and to make easier the technological developments of e-Business systems. A new funded EC project [DBE06] is going to develop platform on which other projects can develop e-Business solutions. Thus, the DBE project contributes to the domain of integration of distributed e-

business solutions. This is targeted also by VISP project but in VISP, the integration is achieved through the design of a common workflow business system.

Standardization in workflow management is more detailed considered and assessed in deliverable D2.1 of VISP project.

2.4. System's and Technological Architecture of Workflow Management Systems

The structure of a workflow management systems consist of a development environment, named build time component and an execution environment, named run time component [zur Muehlen, 2004a]. The development environment gives workflow designers tools for the design of workflow models, the specification of workflow relevant data structures, and the design of the organizational model, relevant to the execution of the workflow models. The modelling of processes and organization structures are mainly supported through graphical modelling tools. The specification of data structures and integration adapters is ordinary text-based and resembles to the traditional programming view. The last can be stored in a workflow model repository.

The second main part of the Workflow management system is the workflow engine or run time environment which consists of modules, which cover different functional aspects. Usually, the modules are hierarchically organized. The coordinator is an event handler, send event notifications to and receive notifications from these modules, [zur Muehlen, 2004a]:

- The *process management facility* create workflow instances from the workflow model. It ensures an appropriate entry to the workflow instance database. It has to obey to the execution constraints like the validity period of a workflow model.
- The *control flow manager* controls state changes of the workflow instances. It has to handle the control flow conditions.
- The *worklist handler* creates and manages access rights to work items. It handles the users' work lists and if conflicts arise, it makes a decision.
- The *user management* facility manages the access of system users to work lists and all the workflow management system. The information from the organizational repository is used by this module for determining of the workflow participants.
- The *application invocation* module controls the interaction of the workflow engine with its applications.
- The *data management* unit executes data conversion and data mapping between activity instances.
- The *history management* component logs system events in the audit trail. These events can be either system related (e. g., user log-on and logoff) or workflow instance related (e. g., activity started, activity completed).
- *Integration APIs* provide access to the workflow engine for external systems. In this way, the workflow engine can be used in another application.

Figure 2.2 shows an example of relationships between the main components in a technical and software system, performing workflow management activities [zur Muehlen, 2004a]:

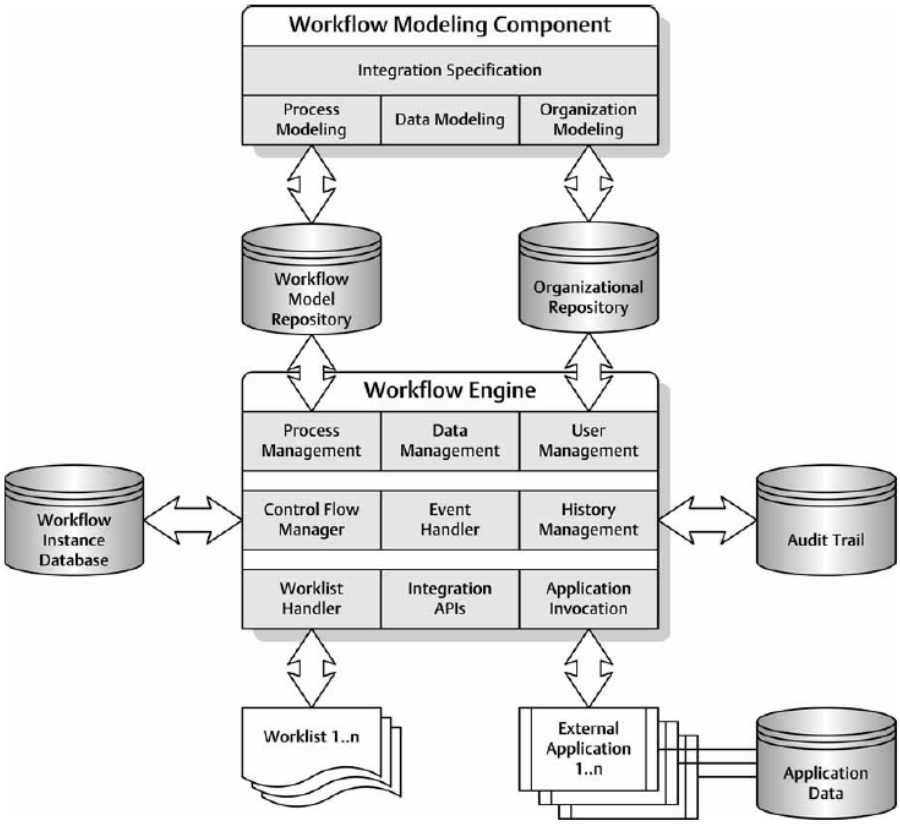


Figure 2.2 - Software architecture of a Workflow Application[2]

2.5. Classes of Workflow Applications

Workflow applications fall into four different categories [Ader, 2004]: production, administrative, collaborative and ad-hoc. The dimensions, along which these kinds of workflow are often described, include:

- repetitiveness and predictability of workflows and tasks;
- how the workflow is initiated and controlled, e.g., from human-controlled to automated;
- requirements for WFMS functionality

Collaborative Workflow involves less rigid processes where the essential feature is providing a structured way for participants to work together. Typical applications are documentation preparation from multiple sources with approval cycles, budget preparation and negotiation or product life cycle management.

[2]Copied from [zur Muehlen, 2004]

Workflow management systems (WFMS) that support collaborative workflows must provide functionality for facilitating human coordination, collaboration, and co-decision [Georgakopoulos, 1995]. Users of a collaborative workflow need to access the WFMS to determine if work was completed. In addition, collaborative WFMSs are not missioning critical, i.e., periodic failure of such workflows does not significantly interfere with the overall business process. Collaborative WFMSs usually use a (proprietary) database to store shared information (e.g., documents such as conference review forms or papers).

The example on Figure 2.3 illustrates a simplified collaborative workflow involving the review process for conference papers. The review process, which starts after collecting all papers, is to select reviewers, distribute the paper(s) to the selected reviewers, collect and reviews and produce a joint review document for each paper, and finally forward it to the authors. This is a collaborative workflow because it involves: (i) negotiation for selecting the reviewers, and (ii) collaboration between the reviewers for producing a joint review. Furthermore, subsequent paper reviews may not be performed by the same reviewers.

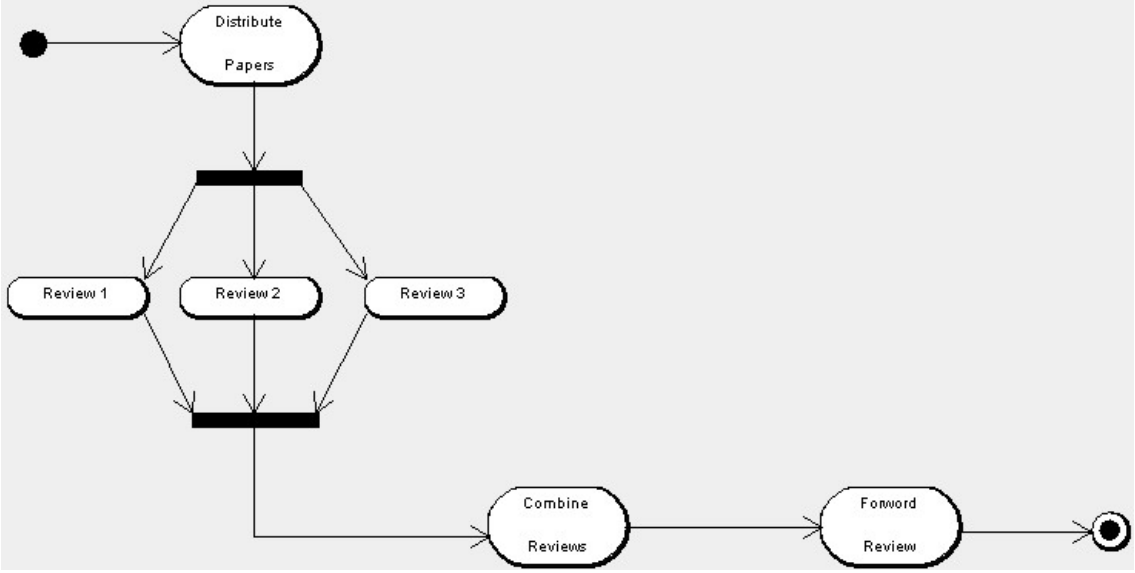


Figure 2.3 - Collaborative workflow

Ad-hoc Workflow relates to applications where the process is not defined in advance and is specific to each process instance. An example is an on-the-fly definition of all the steps and deadlines that are needed to handle a customer query. The critical point here is the ability to define a process in minutes without requiring any help from a specialist. WFMS that support ad hoc workflow are also called *groupware*.

Unlike traditional WFM systems [Cichocki, 1997], in ad-hoc workflow, there is no predefined workflow model for the business process. The business process is built *ad-hocly* as it is enacted using distributed services (activities). Business rules, which capture invariant inter-service logic, are associated with the services and they are used to determine the next activities to be performed on behalf of the business process.

The main advantage of ad-hoc workflow systems is that the business process can be built dynamically and flexible. Another advantage is the ability of ad-hoc workflow to cope with

the dynamic changes in business rules and service definitions, implementations, and/or locations. The service providers taking part in the ad-hoc workflow are not required to be dedicated services. They could be cooperative services in an organization's Intranet, or could be loosely coupled e-commerce services.

The following list outlines the main features necessary for effective ad-hoc and collaborative workflow applications [Ader, 2004]:

- Worklist handler important features: list, order, manage, and select activities.
- History management and graphical status view.
- Activity definition by scripting actions
- Graphical procedure definition
- Features supported by the workflow engine for simple procedure definition
- Activity user agent features: view, manage, enter data, decide and control.
- Dynamic procedure definition change capabilities
- Activity definition through forms designers.
- Libraries of activities and actions

Production Workflow is an intensive application with dedicated agents working full time on repetitive operations. This corresponds to very formal processes with few variations. It requires tight integration with document management applications and legacy systems. Unlike administrative workflow, production workflows typically encompass a complex information process involving access to multiple information systems. The ordering and coordination of tasks in such workflows can be automated. However, automation of production workflows is complicated due to: (i) information process complexity, and (ii) accesses to multiple information systems to perform work and retrieve data for making decisions (administrative workflows rely on humans for most of the decisions and work performed).

WFMS that support production workflow must provide facilities to define task dependencies, and control task execution with little or no human intervention. Production WFMS are often mission critical and must deal with the integration and interoperability of heterogeneous and distributed information systems.

Typical applications are insurance claims processing and loan applications. The illustrative example in Figure 2.4, [Georgakopoulos, 1995] presents a simplified health claims process workflow. In the health claims workflow, a claim form is first manually scanned and stored into an object database. Then the claim is manually indexed in a relational database. This information is subsequently analyzed by an automated "Assess Claim" task. The task is performed by an expert system, which uses an eligibility database to determine if payment should be made. If the claim is rejected, a claim representative discusses the claim with the customer and agrees either to make some payment, or to reject the claim. If payment is made, the "make payment" task accesses the finance database and records the payment. The significant differences between this production workflow and either the ad hoc, collaborative or the administrative workflow is: (i) the interaction of information systems with the business process, and (ii) the use of automated (non-human) task performers.

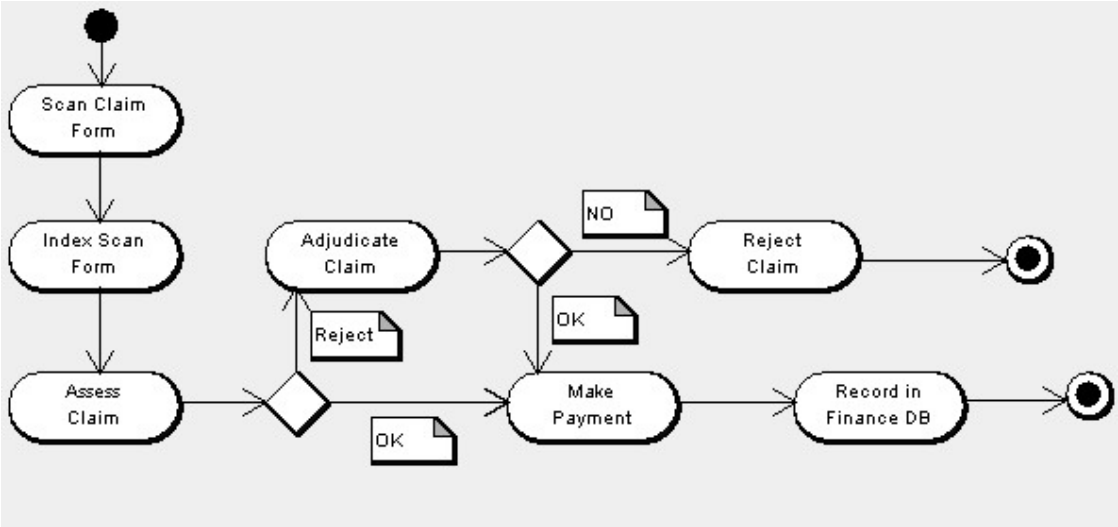


Figure 2.4- Production workflow

Administrative Workflow corresponds to well-defined processes related to day-to-day operations, such as purchase orders, travel requests, expense forms and personnel management processes. It imposes less restrictive performance requirements, uses stable processes and frequently runs with simple forms and document attachments. Many processes can co-exist in the same system, and the number of participants can be very large. A typical application is processing forms. The ordering and coordination of tasks in administrative workflows can be automated. WFMS that support administrative workflow handle simple information routing and document approval functions, such as those found in travel planning and purchase requests. Administrative workflows do not encompass a complex information process and do not require accesses to multiple information system used for supporting production and/or customer services. Administrative WFMS are generally non-mission critical.

Consider again the paper review process assuming that the reviewers are known in advance (e.g., the same reviewers are used for all paper reviews). Furthermore, suppose that the reviewers do not collaborate in producing a joint review. Instead, they produce individual reviews that are considered by the editor (e.g., program chair) who makes the final decision. Under these assumptions, the paper review workflow becomes an administrative workflow such as depicted in Figure 2.5. In an administrative workflow, users are actively *prompted* to perform their tasks. Whereas reviewers using a collaborative workflow needed to access the WFMS to determine if work was completed, reviewers using an administrative WFMS may receive email with review instructions along with the paper to be reviewed and a reviewer's comments form. When the form is completed, it is automatically routed to the program committee chairperson who is alerted when all of the reviews are completed.

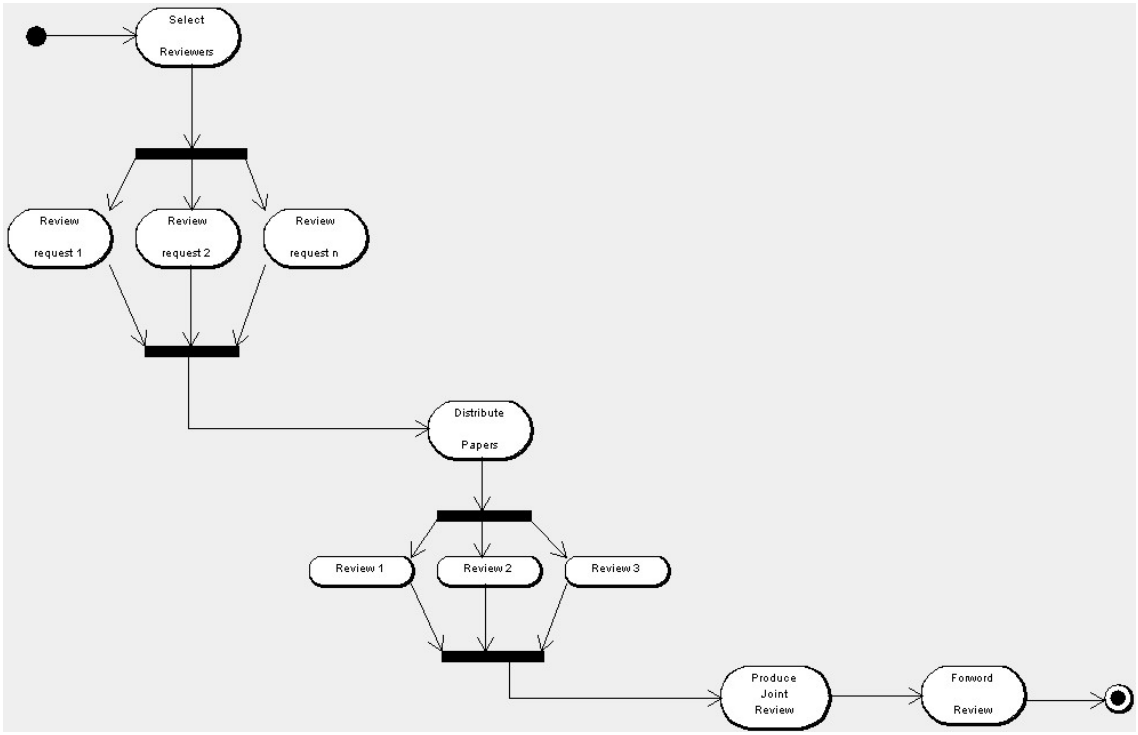


Figure 2.5- Administrative workflow

The following list outlines the main features necessary for effective production and administrative workflow applications [Ader, 2004]:

- Representation of sequence, alternative, parallelism, loop, multiple path
- Variables definition representing the procedure context
- Organization modelling, role, group, participant, authority
- Administration
- Application Programmatic Interfaces
- Multilingual support
- Expression of pre and post conditions, path conditions, exceptions
- Exception processing related to time management
- Verification and simulation
- Dispatching rules based upon an organization model
- Management of deadlines, documents and folders, automatic activities
- Events processing mechanisms, based upon time, and external events.

2.6. Methodology for the Evaluation of Software Tools

The evaluation of a software product can be performed in a different ways. The product quality can be assessed according to quantitative and qualitative software evaluations [Kitchenham, 1996]. The quantitative evaluation identifies the benefits, which are expected from the software tool to deliver in measurable terms. It involves collection of data determining whether the expected results are delivered or not. In [Kitchenham , 1996] three different ways of organizing a quantitative evaluation exercise are defined.

- *Formal experiment*: This is a controlled investigation of an activity, where key factors are identified and manipulated to document their effects on the outcome. Because the formal experiments require a great deal of control, these experiments tend to be small, involving small numbers of people or events [Fenton and Pfleeger, 1996] .
- *Case Study*: This is a research technique where key factors are identified that may affect the outcome of an activity imposed by a real project using the standard project development procedures of the evaluating organization and then document the activity: its inputs, constraints, resources, and outputs. Case studies usually look at a typical project, rather than trying to capture information about all possible cases [Fenton and Pfleeger, 1996].
- *Survey*: A survey is a retrospective study of a situation to try to document relationships and outcomes. A survey is always done after an event has occurred. A survey is a retrospective study by recording situations and comparing them with similar ones. It is important that the investigator does not have control over the situation and the software product. For the case of assessment, it is not possible to manipulate variables, to perform case studies and experiments. Surveys try to mark what is happening broadly over large groups of projects [Fenton and Pfleeger, 1996].

The qualitative methods on their turn assess the extent to which software tools provide the required functionality on a more personal opinion. In [Kitchenham, 1996] the term feature analysis is used to describe such kind of software evaluation. Feature analysis requires a subjective assessment of the relative importance of different features of the products. This kind of analysis is frequently applied different types of weekly and monthly magazines when assessing computer platforms, hardware devices, software tools.

This deliverable categorizes common functionality of available workflow management systems and identifies the scope that these software tools have to cover. This software evaluation applies a combined approach both with quantitative and a qualitative assessments. The deliverable carries out a feature analysis based on documentation of the assessed software tools and when available a first-hand experience of the systems, obtained by software tests. Thus, an attempt to capture the whole functionality of systems was targeted and therefore the evaluation does not focus on a specific application domain. By iteratively comparing the functionality offered by one system with functionality offered by another system it has been identified functionalities common to all of them. The quantitative analysis was performed, using the patterns identified according to the ISO9126 standard for the evaluation of software products. Thus, common frameworks for the assessment of different in nature products were achieved. Due to severe technical problems for the installation of various workflow management systems, this deliverable decided to focus only to prospective software products. The choice of them was done according to internal partner decision, which lacks from theoretical basement and it can be regarded as a subjective point of the evaluation works. Thus, the consortium was able until now to tackle only several systems, which are presented in the short list of Chapter 3 of this deliverable.

For the completeness of the survey, comparisons and available data for other software products have been included. The reasons for constraining the choice are quite practical and they address the possibilities to get an evaluation to actual software versions; the availability of actual components for the software environment; the documentation in common was completely lacking; the software systems had bugs in their normal operation and performance. This is obviously a real situation, which constrained the survey to include more systems.

3. Suites for Workflow Software

Workflow is the operational aspect of a work procedure: how tasks are structured, who performs them, what their relative order is, how they are synchronized, how information flows to support the tasks and how tasks are being tracked [3.1]. As the dimension of time is considered in Workflow, Workflow considers "throughput" as a distinct measure. Workflow problems can be modeled and analyzed using Petri nets [Peterson, 1981], [Reisig, 1992].

While the concept of workflow is not specific to information technology, support for workflow is an integral part of groupware and software. Distinction can be made between "scientific" and "business" workflow paradigms. While the former is mostly concerned with throughput of data through various algorithms, applications and services, the latter concentrates on scheduling task executions, ensuring dependencies that are not necessarily data-driven and may include human agents.

3.1. Classifications of Software Tools

The list of available products for modelling including specification, simulation and execution functionality of the workflow management systems is quite long. During the D2.2 work, a set of 134 software products has been found. This set is not complete, several of the products could be out of date, but they still present in the virtual domain. Thus, the problem of identification and assessment of prospective workflow software products is from high scale and it is difficult to find appropriate solution from users and applications point of view. The products have different level of maturity and they are presented both as market available products, as an open source software suits. Here attempts for classifications of these products are made applying different criteria:

- Classifications of software tools for business and scientific workflows
- Classifications of software tools according to their software language design
- Classifications of software tools according to their supported standard
- Open Source and Commercial Workflow Tools.

3.1.1. Classifications of Software Tools for Business and Scientific Workflows

Workflows systems in scientific domain receive wide acceptance particularly in bioinformatics and cheminformatics in the 2000s. They are considered very successful to perform multiple interconnections between tools, to handle different data formats and data volumes [3.1]. The models of scientific workflows are close related to the traditional CGI and server language scripting. The adoption of the workflow methodology represents a natural step towards the implementation of automatic techniques in researches and experimental data processing and retrieval.

The workflow systems in the business domain are more generic. They represent in a structural way the set of business tasks in the organizations, perform the times scheduling between the processes, coordinate software applications, manage the paper flow documentation within an organization. This system origins since 1970s, when the organization were paper-based. Later the workflow management moves to modern computers and IT systems. As a way of bridging the gap between the two paper-based and paperless

technologies, significant effort is being put into defining workflow patterns that can be used to compare and contrast different workflow engines across both of these domains.

Workflow systems are defined as "systems that help organizations to specify, execute, monitor, and coordinate the flow of work cases within a distributed office environment" [3.1]. [3.1] http://en.wikipedia.org/wiki/WorkflowThe Workflow diagrams rely on the use of standardized graphical notations to describe workflow structures where Business Process Modelling Notation is an example of this system. The system contains two basic components:

- The first component is the workflow modelling component (sometimes called the specification module or the build time system), which enables administrations and analysts to define process and activities, analyze and simulate them, and assign them to people.
- The second component is the workflow execution component, sometimes called the run-time system. It consists of the execution interface seen by end-users and the workflow engine, an execution environment that assists in coordination and performing the processes and activities.

Applying this categorization of the software tools for business and scientific workflow systems, an available list of products [3.2] is the following:

- Tools for business workflow

o @enterprise http://www.groiss.com/en/index.html	o Aegeanet System http://www.aegeanet-system.com/
o Agentflow http://www.flowring.com/flowring/en_page/index.jsp	o Amazonas Worflow http://objeng.ch/web-oe/sw-products/awf/awf-de.html
o Bonita http://bonita.objectweb.org	o Business Integration Engine http://www.brunswickwdi.com/what_is_bie
o Business Process Management http://www.cosa.de/BPM_Produkte.html	o Captaris http://www.captaris.com/workflow/
o COLOSA http://www.colosa.com/	o EmeriCon http://www.emericon.com/index.html
o Enhydra Shark http://shark.objectweb.org	o EventStudio http://www.eventhelix.com/EventStudio/
o FlowRunner http://www.brightsoft.com/	o iKE http://www.ike.com/
o Ils/process http://www.ilogs.com/en/html/index.php	o infoRouter http://www.inforouter.com/
o IngTech Corporation http://www.ing-tech.com/	o Interstage BPM http://www.fujitsu.com/global/services/software/interstage/products/bpm/
o Jboss jBPM http://jbpm.org	o K2.net Enterprise Workflow http://www.k2workflow.com/Default.aspx
o MyControl Workflow Server http://www.mycontrol.de/workflow/	o OpenFlow http://www.openflow.it
o OpenSymphony http://www.opensymphony.com/osworkflow	o OpenWFE http://web.openwfe.org/display/openwfe/Home
o Oracle BPEL Process Manager	o PL/FLOW

http://www.oracle.com/technology/products/ias/bpel/index.html	http://plflow.sourceforge.net/
○ Skelta Workflow.NET http://www.skelta.com/	○ VivTek http://www.vivtek.com/wftk/
○ W4 http://www.w4global.com/	○ Web and Flo Kontinuum http://www.webandflo.com/
○ WebSphere MQ Workflow http://www-306.ibm.com/software/integration/wmqwf/	○ YAWL http://www.yawl.fit.qut.edu.au/

- Tools for scientific workflow

○ Taverna http://taverna.sourceforge.net/	○ Kepler http://kepler-project.org/
○ GridNexus http://www.gridnexus.org/	○ SPA http://www-casc.llnl.gov/sdm/
○ Triana http://www.trianacode.org/	○ JOpera http://www.jopera.ethz.ch/

3.1.2. Classifications Of Software Tools According To Their Software Language Design

A review of the set of open source workflow project is given in [3.3]. A classification is performed according to program language, used for the design of the software tool. The categories used in this classification are projects, written in Java; other Java; other non-Java; for specific application servers or environment.

- Open Source Workflow Engines Written in Java.

The review list consists:

• ActiveBPEL	• Antflow
• Apache Agila	• Beexee
• Bigbross Bossa	• Codehaus Werkflow Werkflow
• con:cern	• Dalma
• Enhydra Shark	• Freefluo
• jBpm	• Jfolder
• MidOffice BPEL Engine	• Micro-Workflow
• ObjectWeb Bonita	• OFBiz
• OpenWFE	• OpenSymphony OSWorkflow
• Pi Calculus for SOA	• PXE
• Syrup	• Taverna
• Twister	• wfmOpen
• Xflow2	• YAWL
• Zbuilder	• Zebra

An extended description of each product is available from [3.4].

A list with prospective tools is given below.

- Generic J2EE

• Apache Agila	• Bonita

• Imixs	• Jboss jBpm
• Jfolder	• Open Business Engine
• wfmOpen	• XFLow

- Other Java based products

• Bossa	• Enhydra Shark
• Open Symphony Workflow (OSWorkflow)	• OpenWFE
• Syrup	• Twister
• Werkflow	• Yet Another Workflow Language engine
• Zebra	•

- Other: Non-Java tools

• wftk (workflow toolkit)	•

- Other based: For specific application servers or environment

• Galaxia Tikiwiki	• Grid-based computing workflow – Taverna
• Open for Business (OfBiz)	• OpenFlow
• webMethods Business Process Management	• BizFlow
• ActionWorks	• WebAsyst Workflow Management Software
• WebAsyst Issue Tracking	

3.1.3. Classifications Of Software Tools According To Their Supported Standard

A short classification, performed according to the supported workflow standards and business modelling is done below.

- XPDL based

• Aspose.Workflow	• Vignette Process Workflow Modeler
• Agentflow	• Aspose Workflow
• Enhydra JaWE	• ObjectWeb BONITA
• Enhydra Shark	• Fuego BPM
• Interstage BPM	• Newgen
• OFBiz Workflow Engine	• Open Business Engine
• wfmOpen	• YAWL

- BPEL based

• iGrafx BPEL	• Apache Agila
• ActiveBPEL Engine	• Bexee
• ActiveWebflow Standard	• Cape Clear

• Biztalk Server	• IBM BPWS4J
• Oracle BPEL Process Manager	• JOpera
• PXE	• MidOffice
• Parasoft BPEL Maestro	• Twister
• SAP NetWeaver Exchange Infrastructure	

- BPMN based

• Borland Together Desidner 2006	• ITpearls Process Modeler
• IntalioDesigner	• AXway Process Manager™
• Fujitsu: Interstage Business Process Manager 7.1	• Kaisha-Tec: ActiveModeler
• Lanner: Witness™	• Mega International: Mega Suite™

3.1.4. Open Source and Commercial Workflow Tools

- Open Source Workflow Tools

This classification concerns the software products, which are offered as open code products and do not belong to propriety constraints. The characteristics of the products are described according to the data available from the corresponding web addresses and web sources.

• Twister	• jBPM
• Enhydra Shark	• OpenSymphony OSWorkflow
• con:cern	• Codehaus Werkflow
• ObjectWeb Bonita	• Bigbross Bossa
• Open Business Engine	• The Open for Business Workflow Engine
• OpenWFE	• WfMOpen
• XFlow	• Jfolder
• Taverna	• Freefluo
• Micro-Flow	• Jflower
• YAWL	• Syrup
• PXE	• ActiveBPEL
• Antflow	• Swish

Additional information can be found also in [3.5].

- Commercial Workflow Tools

• Active Endpoints ActiveWebflow Server	• ActiveWebflow Designer
• ADONIS	• Biztalk Server
• Cape Clear Orchestrator	• Digité Process Composer
• Fiorano SOA Platform	• FiveSight PXE
• FuegoBPM	• IBM BPWS4J

• IBM WebSphere Business Integration Server Foundation	• OpenLink Virtuoso Universal Server
• OpenStorm ChoreoServer	• Oracle BPEL Process Manager
• Parasoft BPEL Maestro	• PolarLake Integration Suite
• SAP NetWeaver Exchange Infrastructure	• SeeBeyond eInsight BPM

- Commercial vendors
  - Bea's WLI [3.6]
  - Carnot [3.7]
  - Dralasoft [3.8]
  - Filenet [3.9]
  - Fujitsu's i-Flow [3.10]
  - IBM's holosofx tool [3.11]
  - Intalio [3.12]
  - Lombardi [3.13]
  - Oakgrove's reactor [3.14]
  - Oracle's integration platform [3.15]
  - Q-Link [3.16]
  - SAP's NetWeaver [3.17]
  - Savvion [3.18]
  - Seebeyond [3.19]
  - Sonic's orchestration server [3.20]
  - Staffware [3.21]
  - Ultimus [3.22]
  - Versata [3.23]
  - WebMethod's process modelling [3.24]

- Tool catalogs
  - http://www.telelogic.com/corp/
  - http://dmoz.org/Computers/Software/Workflow/Products/
  - http://java-source.net/open-source/workflow-engines

The survey of products, available for the domain of workflow modelling, specification and execution describe a huge amount of products. It is difficult for a common user to make appropriate decisions about his choice for taking and working with appropriate software tool. The problem of evaluation and assessing this class of software has been attack from different point of views. In the next paragraph, a comparative analysis is given about the current state of evaluation of workflow products and software tools.

3.2. Current Experience in Evaluation of Workflow Software Products

3.2.1. Comparisons And Assessments Of Workflow Software Products And Tools

The workflow management systems (or business process management systems) developed in the late 1980s enable the automated coordination of activities, data and resources. This

coordination is formalized in a workflow model specified at the build-time stage [Georgakopoulos *et al*., 1995]. The problem of evaluation of the quality of the software product covering different functionalities of workflow management system is currently attacked by the research and technological community. An attempt for obtaining practical and useful results in the product assessment is currently made in [BPEL-UniS]. These attempts prove the importance of finding practical solutions for the evaluation of software products in this hot topic-workflow management.

The architecture of workflow management systems can be separated in a development environment (modelling with build time component) and an execution environment (run time engine component) [zur Muehlen, 2004]. Both these two general components of the software tools for workflow management are considered below.

An evaluation of several software products has been done in [Haller *et al*., 2005]. This work is essential because it illustrates the current practice, the maturity and the usefulness of the applied methodology for comparison of the workflow software product. Thus, it can be proved that the problem of evaluation of workflow product is not a trivial technological problem. Even the results obtained from such a comparison cannot give benefits for direct utilization from user point of view. For this practical attempt to evaluate workflow software, the products under considerations are i-Flow Interstage Business platform of the company Fujitsu Regatta and Websphere MQ Workflow, which is part of the Websphere MQ product line of IBM. The comparison addresses five general aspects of evaluation: functional, behavioural, operational, informational and organizational.

- The *functional* aspect presents a functional decomposition of activities in the workflow network.
- The *behavioural* aspect focuses on the time. It describes the execution order and dependencies of the control flows and activities in the workflow.
- The *operational* aspect describes the interaction of the workflow management system with its environment, the methods for accessing or invoking external applications (e.g. interaction mode, invocation mode, parameters) and the communication with human users;
- The *informational* aspect describes the data exchange between activities. The data in a workflow management system can be distinguished to internal data (managed by the workflow management system) and external data (exist even without workflow and might be used);
- The *organizational* aspect addresses the resources executing the work. The resources can be human employees, computers, additions necessary for the specific workflow, etc. The organization can have hierarchical structure and resource allocation to activities has to be done.

Qualitative Analysis

i-Flow Interstage Business platform

i-Flow Interstage Business platform is a Business Process Management. The system has 4-tier architecture: repository, process logic, web tier, user interface [Haller *et al*., 2005]. The web tier contains servlet components running in a web server. The main process logic is in the business process management tier. The fourth tier contains the database directory, document management as well as connectivity to other systems. The main tool within the Interstage suite is the Development Manager Client. It defines workflow types (called

workflow templates in i-Flow) and manages the execution of workflows. The Administration Client has the responsibility for administering processes and templates.

A. *Functional aspect*. The tasks, performed in the process, the operators representing the workflow activities are components of i-Flow workflow templates. The workflow template has one start node and at least one end node. Workflow templates can be organized in super and respective sub-workflows.

B. *Behavioural aspect*. i-Flow provides a graphical user interface to create process templates. The units below are available for designing the sequential behaviour

- *Arrow*: They are used to link two activities to denote the flow of events.
- *Conditional* Node: This node realizes an exclusive OR or an AND. A default outgoing arrow must be chosen to allow process ending if conditions are not satisfied. The order for conditions' evaluation can be determined. For conditions' defining are applied standard JavaScript expressions.
- *Complex Conditional* Node: The behaviour of this node is similar to the Conditional Node. Here the expressions for defining conditions can include multiple incoming data items.
- *OR* Node: It can be used either as a split construct (it behaves as a parallel split that executes all subsequent activities simultaneously or in any order), or as a join construct (the OR node merges two or more branches without synchronization: the subsequent activity is started as soon as one branch is finished, but when the subsequent activity finishes all active branches are terminated).
- *AND* Node: It can be used as a split construct (it is identical to the behaviour of the OR node). When it is a join construct, it is synchronizing: the process waits until all activities that lead to this node are completed, before starting the subsequent activity.
- *Subprocess* Node: The workflow reaching this node set process control to a sub-process, and the node enters a Waiting-for-Sub-process state. When the sub-process completes, control returns to this node, and all ensuing activities connected to the node are activated simultaneously. Both run-time and design-time sub-processes are used to break complex tasks into a hierarchy of easier-to-handle units.
- *Remote Sub-process* Node: This node corresponds to the sub-process node with the only difference that the sub-process resides in another workflow engine. i-Flow supports other i-Flow servers, Collaboration Ring or other SWAP compatible process engines. So that it is possible to start a remote sub-process from the local i-Flow Server. The local process waits for the remote process to complete, and incorporates the results of the remote process back into the local process.
- *Chained Process* Node: When the workflow reaches this node, another process is activated similarly to a sub-process. The difference is that the node does not enter a suspended state, and the chained process operates as an independent entity.
- *Delay* Node: When the workflow reaches this node, the process pauses until the first timer attached to the node fires.
- *Triggers*: Triggers either start processes or make choices of activities in response to Data Event files (added to a particular directory by a data source external to i-Flow).
- *Exit* Node: Identifies the end of the process. Every process must have at least one Exit Node.

C. *Informational aspect.* i-Flow uses the principle of globally shared data. All data items defined within a workflow type are shared between the tasks. Only between super and sub-

workflows a distinct data mapping has to be defined to pass data elements back and forth. External data passed to i-Flow in the prologue or epilogue actions of any node are always mapped to so call User-defined process attributes (UDAs). UDAs are variables, which hold values in a running process, set by a user with a form, script, or a JavaAction. Triggers allow mapping external data from XML files to UDAs. By loading a XML schema, file mappings can be defined from each element to define UDAs for the activity.

D. *Operational aspect.* i-Flow provides communication with external applications over Java Actions. A Java Action is a call to a static method on a specified Java class. The Java action is configured to pass UDA values as parameters, and the result of the method is assigned to a UDA. A Java Action can be invoked when a process starts, when a node is activated or completed, when a timer expires, at time of role resolution and when the process ends. Many nodes (see table 3.1) have the ability to specify a list of actions to be performed when activated.

Table 3.1: List of nodes and their perspective support for Java Actions

Node	Prologue	Epilogue
OR Node		yes
AND Node		yes
Condition Node	yes	
Start Node		
Stop Node		
Chained Subprocess	yes	yes
Activity	yes	yes
Voting Activity	yes	yes
Subprocess	yes	yes
Remote Subprocess	yes	yes
Delay	yes	yes

It is distinguished between Prologue Actions performed in a node before all its other assigned activities are executed and epilogue Java Actions performed as closure actions in a node. Java Actions allow executing Java business methods that are outside the scope of i-Flow. By using Generic Java Actions, it is possible to use methods within autonomous Java classes. A Java Action can pass an Enactment context to the invoked application, which constitutes an API that can be used to retrieve information out of the process instance.
Action Sets can also execute rules. Rules have to be defined in a rules file created with a supported rules engine. The rules engine must have access to the i-Flow Business Object Model used by i-Flow to implement the rules created by the rules engine.

E. *Organizational aspect* At design time, template owners have the option of assigning process ownership. Ownership can be assigned to any i-Flow role or set of users. Activities are assigned to a role, a user, or a set of users. If more than one user is assigned, each assignee receives a separate activity in the Activity List. Every user has its own profile, defining the way the user is notified when a work-item is assigned and the association of a DMS directory for its user ID. The reassignment Java action allows reassigning tasks. A task can either be manually reassigned or automatically to a user or users with the predefined "Assign Task to User Action". A work-item owner can also reassign the task to any other role, whereas the new owner can accept or reject the reassignment. The directory adapter

implements an i-Flow specific interface to expand a group into a list of individuals. The enactment engine uses this at runtime to determine work assignments. The ability to read the directory or to expand groups into a list of individuals is exposed by the Model API. i-Flow provides adaptors to standard LDAP directory such as Sun ONE Directory Server, to Microsoft Active Directory and to Microsoft Windows native user/group support.
By the same way have been described the second product Websphere MQ Workflow, which is used for the comparison with i-Flow Interstage Business platform. This descriptive way is worth for experts in the workflow domain but it does not give relative knowledge about the quality of these two products. The evaluation continues with the introduction of quantitative analysis, which has to introduce appropriate score scheme for the product assessment. For illustration purposes, the qualitative analysis is referred by [Haller *et al*., 2005].

Quantitative Relative Analysis of the products iFlow and Websphere MQ Workflow

The quantitative analysis describes the level of support in each workflow management systems for the control flow and data patterns. These patterns are summarized and then they have been evaluated for each pattern if it is supported in each workflow management system. The control patterns are divided in several groups: basic patterns advanced branching patterns, structural patterns, multiple instance patterns, state-based patterns, and cancellation patterns [Haller *et al*., 2005].

The basic patterns capture elementary control flow consists: a *sequence* of activities in which two activities are executed after each other. It means that the second one becomes enabled when the first one completes; a *parallel split* and *parallel synchronization (and-join)* of activities in which several activities execute in parallel (or in any order). Later they are synchronized (the synchronize activity waits until all parallel branches are completed, and then continues). An *exclusive choice (xor-split)* and *exclusive synchronization (xor-join)* in which one out of several branches is chosen based on some condition and later synchronized (the synchronize activity waits until the active branch completes, since only one branch is assumed to be active).

The results of the comparisons are expressed in a graphical way. For the case of control flow patterns here are presented the results of the comparison in [Haller *et al*., 2005] for implementation of Pattern1 (sequence), Pattern 2 (Parallel split), Pattern 3 (Synchronization), and Pattern 4 (Exclusive choice).

Pattern 1 (Sequence): Two or more activities in a workflow type are processed in the order, in which they were defined. This most basic construct is achieved in i-Flow by connecting two activities with an arrow, Figure 3.1. In MQ Workflow, this construct is achieved by connecting two activities with a control connector, Figure 3.2.

Pattern 2 (Parallel split): This pattern defines a point in a process, which is split into several threads of control, all executed in parallel, and whereas the order in which they are processed is not defined. In i-Flow this pattern can be achieved by using the AND Node as shown in Figure 3.3. In MQ Workflow this construct is achieved by having multiple outgoing control connectors from one activity: if the control connectors do not contain transition conditions they are by default all enabled, which means that all branches will be activated, Figure 3.4. In MQ Workflow no explicit control flow nodes exist, instead control flow constructs are implemented, using transition conditions on the control connectors (arrows).

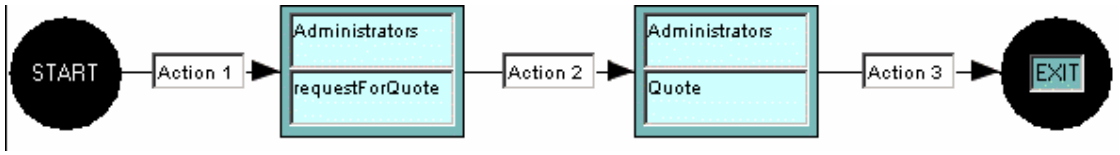


Figure3.1 - Workflow Pattern 1 (Sequence) in i-Flow

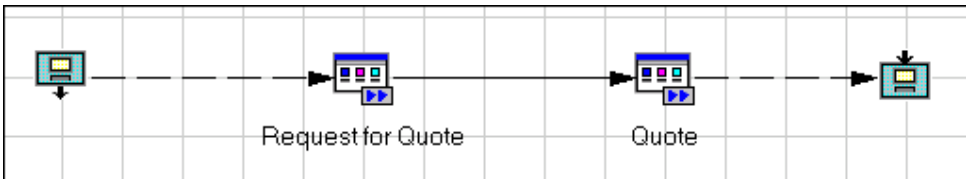


Figure3.2 - Workflow Pattern 1 (Sequence) in MQ Workflow

Pattern 3 (Synchronization): A point where multiple parallel sub-processes converge into one thread, whereas each incoming branch is executed only once. In i-Flow the AND node is used to merge any number of incoming branches. The process will only continue if every preceding activity is finished, Figure 3.5. In MQ Workflow, no explicit synchronization node is available or necessary; this construct is achieved by having multiple incoming control connectors to an activity. Multiple incoming control connectors mean that the activity is only enabled if all incoming control connectors are enabled, which means that all preceding activities have terminated, Figure 3.6.

Pattern 4 (Exclusive choice): It defines a point in the process where a certain condition based on a decision in the flow is taken. I-Flow does not support an explicit XOR construct, but a conditional node to define exclusive transition conditions, Figure 3.7. In MQ Workflow, a similar approach must be taken: a transition condition on the control connector defines which branches are activated; setting excluding transition conditions on two branches ensure exclusive-choice behaviour. Transition conditions are not graphically visible, Figure 3.8.

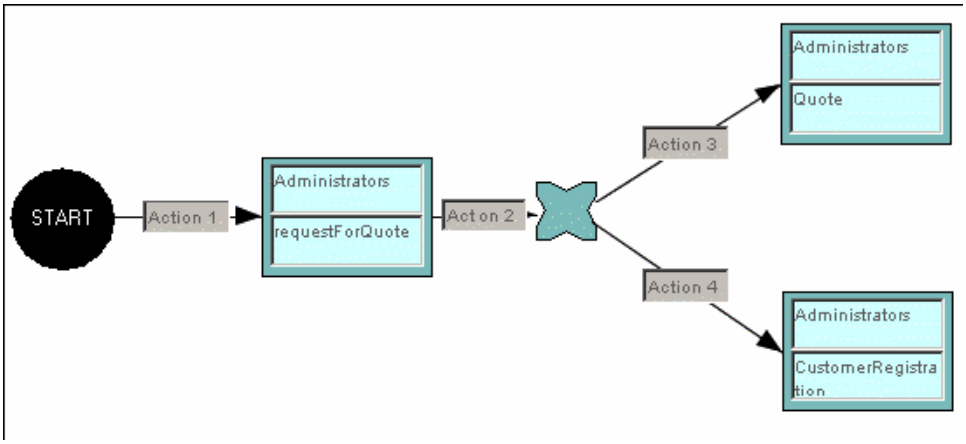


Figure3.3 - Workflow Pattern 2 (Parallel Split) in i-Flow

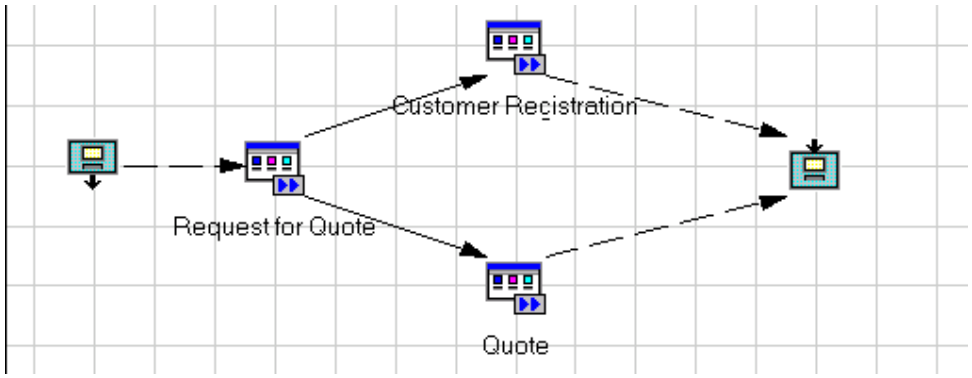


Figure3.4 - Workflow Pattern 2 (Parallel Split) in MQ Workflow

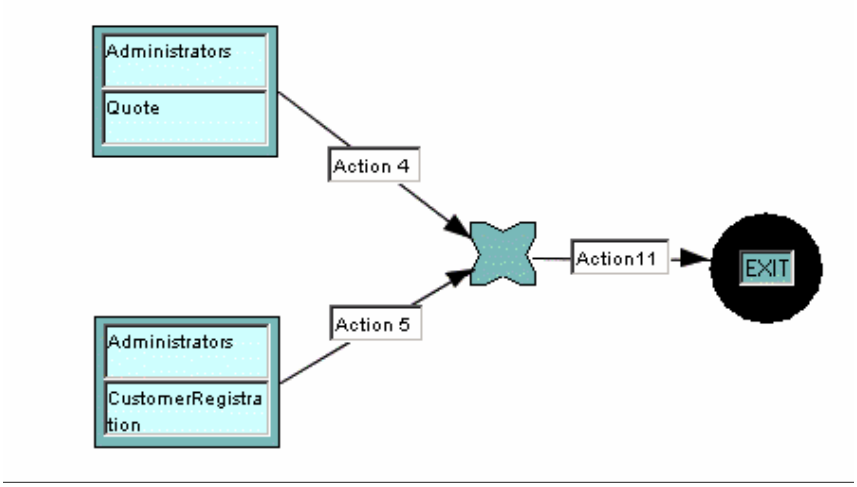


Figure3.5 - Workflow Pattern 3 (Synchronization) in i-Flow

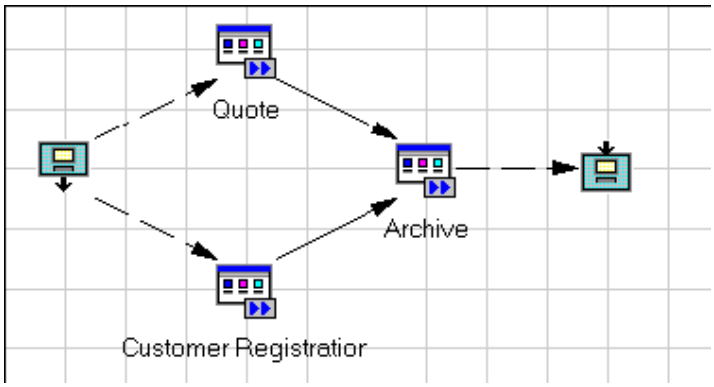


Figure3.6 - Workflow Pattern 3 (Synchronization) in MQ Workflow

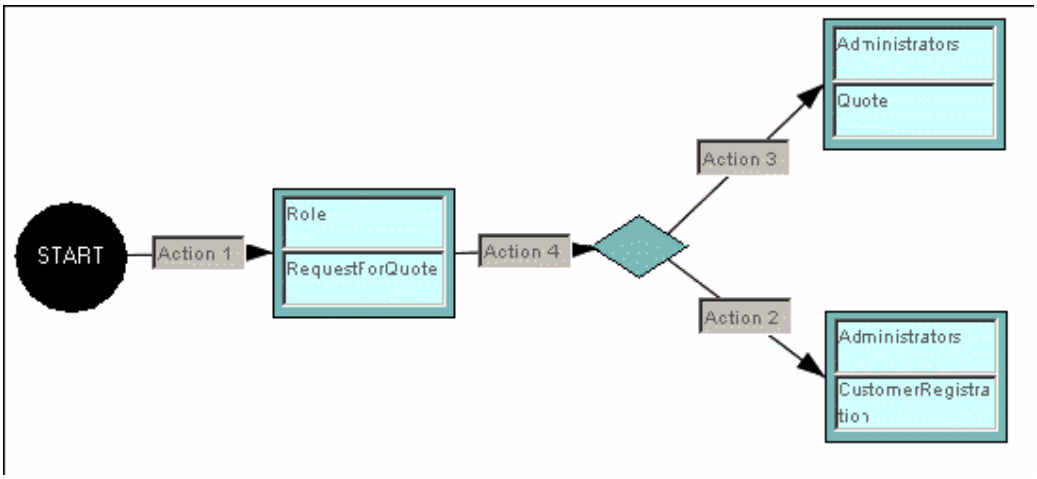


Figure3.7 - Workflow Pattern 4 (Exclusive choice) in i-Flow

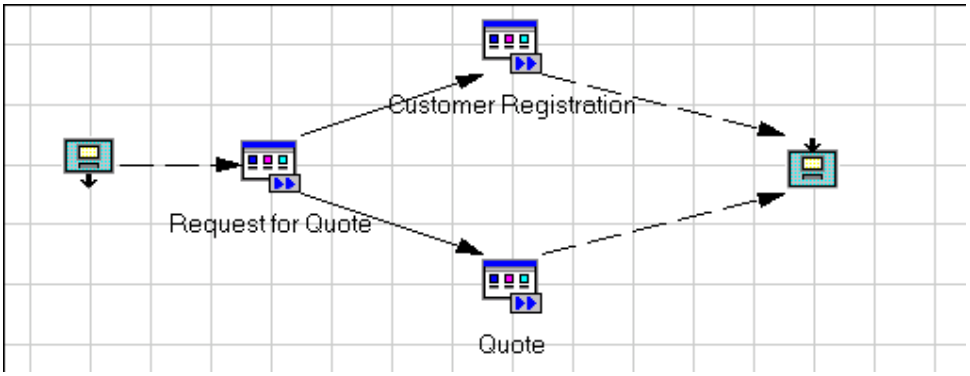


Figure3.8 - Workflow Pattern 4 (Exclusive choice) in MQ Workflow

Despite that, the evaluation is titled "Quantitative Relative Analysis" gives graphical representation of the tools about their implementation of workflow patterns. Such graphical comparison does not introduce quantitative evaluation scheme and particularly the user cannot make right decision about the quality of the products. The derived practical conclusions for user utilization are quite few. This is a result both from the complexity of the evaluated tools as from the lack of common evaluation methodology, which can derive practical and useful conclusions.

3.2.2. Evaluation of Workflow Software Products According To the Workflow Patterns Supported

The workflow patterns initiative started at the Fourth IFCIS International Conference on Cooperative Information Systems in Edinburgh in 1999 [Aalst et al., 2000]. In the beginning of 2001 when the workflow patterns site [3.25] was released, it attracts the attention of researchers and practitioners to apply the patterns in product development and product evaluations. The workflow patterns become an evaluation framework for the assessment of

the quality of the workflow software products. The results of the patterns evaluations were applied to ERP-systems, E-business systems, Call-center software. An assessment study, addressing the quality of workflow software products based on pattern evaluation, was performed in [Aalst *et al*., 2003]. To illustrate this methodology and its evaluation results here is presented the sequence and part of the results, obtained for the product evaluation.
Some of the patterns, which have been used for the comparative study and which match the definitions of elementary control flow concepts, are:
Pattern 1 (Sequence); **Pattern 2** (Parallel Split); **Pattern 3** (Synchronization; **Pattern 4** (Exclusive Choice); **Pattern 5** (Simple Merge);**Pattern 6** (Multi-choice), Figure 3.9; **Pattern 7** (Synchronizing Merge);

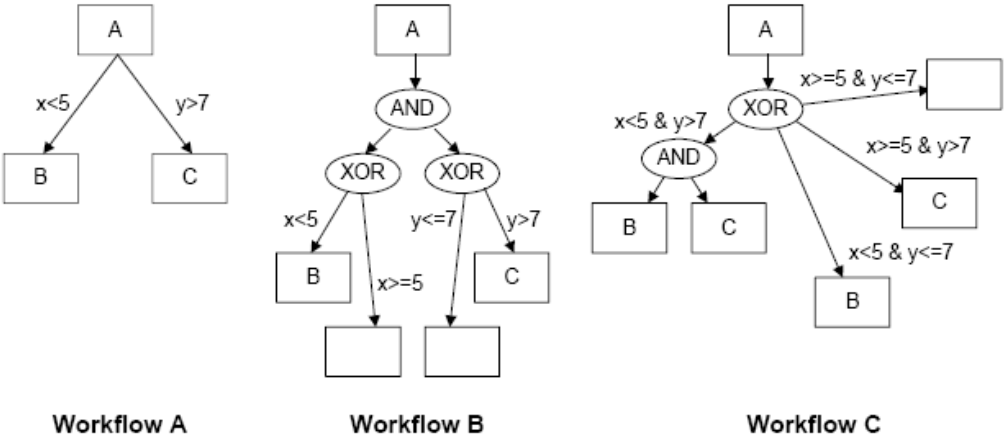


Figure3.9 - Design patterns for the multi-choice

Pattern 8 (Multi-merge); **Pattern 9** (Discriminator); **Pattern 10** (Arbitrary Cycles); **Pattern 11** (Implicit Termination); **Pattern 12** (Multiple Instances without Synchronization); **Pattern 13** (Multiple Instances with a Priori Design Time Knowledge); **Pattern 14** (Multiple Instances with a Priori Runtime Knowledge); **Pattern 15** (Multiple Instances without a Priori Runtime Knowledge); **Pattern 16** (Deferred Choice); **Pattern 17** (Interleaved Parallel Routing); **Pattern 18** (Milestone); **Pattern 19** (Cancel Activity); **Pattern 20** (Cancel Case).

The workflow patterns correspond to routing constructs encountered when modelling and analyzing workflows by software products of workflow management systems. However, several patterns are difficult to realize using many of the workflow management systems available today. Thus the pattern functionality can be into account when compare son and evaluation is performed for software workflow products. The products can be checked for the presence of sequential, parallel, conditional, and iterative routing.

In [Aalst et al, 2002] a comparison of the functionality of 15 workflow management systems using the workflow patterns as evaluation criteria is performed. The evaluated products are:

- **COSA** is a Petri-net-based workflow management system developed by Ley GmbH, formerly operating under the names Software Ley, COSA Solutions, and Baan).
- **FLOWer** is Pallas Athena's case handling product.
- **DominoWorkow** is the workflow extension of the widely used groupware product Lotus Domino/Notes (Lotus/IBM).

- **Eastman Software** offers a variety of imaging products. Their software is used to electronically capture, share, display, fax, print, and store vital document-based information. On top of their imaging products, Eastman Software also offers a workflow management system.
- **Visual WorkFlow** is one of the market leaders in the workflow industry. It is part of the **FileNet's Panagon** suite (Panagon WorkFlo Services) that includes also document management and imaging servers.
- **Forté Conductor** is a workflow engine that is an add-on to Forte's development environment, Forte 4GL (formerly Forte Application Environment).
- **Meteor** (Managing End-To-End OpeRations) is a CORBA-based workflow management system developed by members of the LSDIS laboratory of the University of Georgia (USA).
- **Mobile** is a workflow management system developed by members of the Database Systems group at the University of Erlangen/Nurnberg (Germany).
- **MQSeries/Workflow** is the successor of IBM's workflow offering, FlowMark.
- **Staffware** is one of the leading workflow management systems. Staffware is authored and distributed by Staffware PLC.
- **Verve** debuted to the workflow market as in 1998. In late 2000 it was acquired by Versata and renamed Versata Integration Server (VIS).
- **I-Flow** is a workflow offering from Fujitsu that can be seen as a successor of the workflow engine from the same company, TeamWare. I-Flow is web-centric and has a Java/CORBA based engine built specifically for Independent Software Vendors and System Integrators.
- **Changengine** is a workflow offering from HP, the second largest computer supplier in the world.
- **SAP R/3 Workflow** SAP is the main player in the market of ERP systems.

For each product-pattern combination was checked whether it is possible to realize the workflow pattern with the tool. If a product directly supports the pattern through one of its constructs, it is rated "**+**". If the pattern is not directly supported, it is rated "**+/-**". Any solution which doesn't give direct support is rated "**-**". The two tables 3.2 and 3.3 summarize the evaluation results of the products:

Table 3.2 Product Pattern

pattern	*products*							
	Staffware	COSA	InConcert	Eastman	FLOWer	Domino	Meteor	Mobile
Sequence	+	+	+	+	+	+	+	+
Parallel Split	+	+	+	+	+	+	+	+
Synchronization	+	+	+	+	+	+	+	+
Exclusive Choice	+	+	+/-	+	+	+	+	+
Simple Merge	+	+	+/-	+	+	+	+	+
Multi Choice	-	+	+/-	+/-	-	+	+	+
Synchronizing Merge	-	+/-	+	+	-	+	-	-
Multi Merge	-	-	-	+	+/-	+/-	+	-
Discriminator	-	-	-	+	+/-	-	+/-	+
Arbitrary Cycles	+	+	-	+	-	+	+	-
Implicit Termination	+	-	+	+	-	+	-	-
MI without Synchronization	-	+/-	-	+	+	+/-	+	-

MI with a Priori Design Time Knowledge	+	+	+	+	+	+	+	+
MI with a Priori Runtime Knowledge	-	-	-	-	+	-	-	-
MI without a Priori Runtime Knowledge	-	-	-	-	+	-	-	-
Deferred Choice	-	+	-	-	+/-	-	-	-
Interleaved Parallel Routing	-	+	-	-	+/-	-	-	+
Milestone	-	+	-	-	+/-	-	-	-
Cancel Activity	+	+	-	-	+/-	-	-	-
Cancel Case	-	-	-	-	+/-	+	-	-

Table 3.3 Product Pattern

pattern	*product*						
	MQSeries	Forté	Verve	Vis. WF	Changemg.	I_Flow	SAP/R3
Sequence	+	+	+	+	+	+	+
Parallel Split	+	+	+	+	+	+	+
Synchronization	+	+	+	+	+	+	+
Exclusive Choice	+	+	+	+	+	+	+
Simple Merge	+	+	+	+	+	+	+
Multi Choice	+	+	+	+	+	+	+
Synchronizing Merge	+	-	-	-	-	-	-
Multi Merge	-	+	+	-	-	-	-
Discriminator	-	+	+	-	+	-	+
Arbitrary Cycles	-	+	+	+/-	+	+	-
Implicit Termination	+	-	-	-	-	-	-
MI without Synchronization	-	+	+	+	-	+	-
MI with a Priori Design Time Knowledge	+	+	+	+	+	+	+
MI with a Priori Runtime Knowledge	+/-	-	-	-	-	-	+/-
MI without a Priori Runtime Knowledge	-	-	-	-	-	-	-
Deferred Choice	-	-	-	-	-	-	-
Interleaved Parallel Routing	-	-	-	-	-	-	-
Milestone	-	-	-	-	-	-	-
Cancel Activity	-	-	-	-	-	-	+
Cancel Case	-	+	+	-	+	-	+

From the comparison it is clear that none of the tools support all the selected patterns. In fact, many of these tools only support a relatively small subset of the more advanced patterns.
The assessment of these products has been extended in [Kiepuszewski, 2003]. It has been graphically illustrated how these products implement a sample process shown in Figure 3.10 [Kiepuszewski, 2003]. [Kiepuszewski, 2003] . Kiepuszewski, B. (2003) Expressiveness and

Suitability ofThis process employs all WfMC-defined basic constructs, which are used in a typical context leaving no scope for ambiguous interpretation.

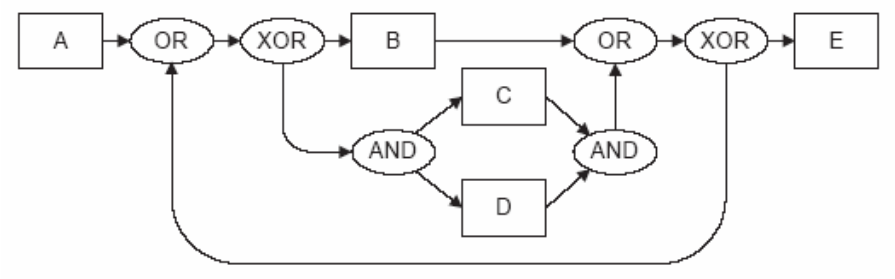


Figure 3.10 - Sample process model

FileNet's Visual WorkFlo

Visual WorkFlo [Fil97], [Fil99] is one of the leading workflow products in the industry. It is part of the FileNet's Panagon suite (PanagonWorkFlo Services), which includes document management and imaging servers as well. Visual WorkFlo is one of the oldest and best-established products on the market. Since its introduction in 1994, it managed to gain a respectable share of all worldwide workflow applications. The evaluation here is based on version 3.0 (introduced in late 1998) of the product.
Figure 3.11 shows the sample process introduced in Figure 3.10 implemented with Visual WorkFlo.

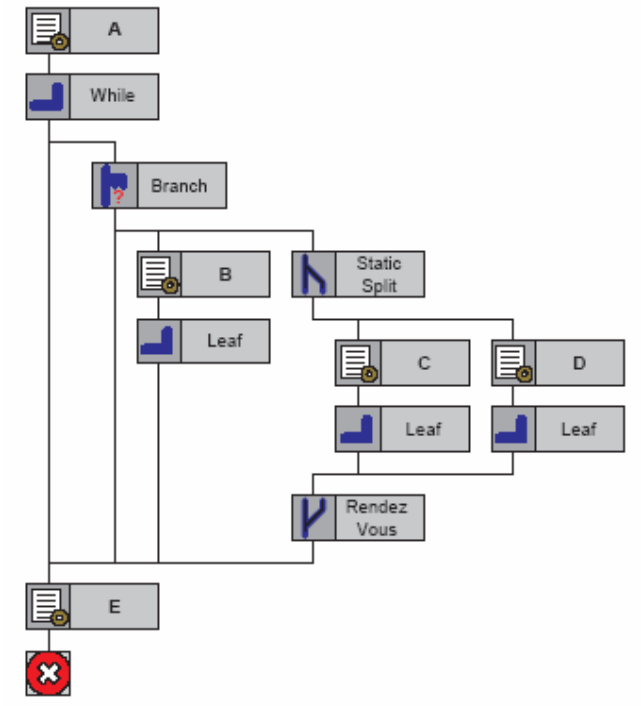


Figure3.11 - Sample process model implemented with Visual WorkFlo

Forté Conductor

Forté Conductor [For98] is a workflow engine that is an add-on to Forté's development environment, Forté 4GL (formerly Forté Application Environment). Conductor's engine is based on experimental work performed at Digital Research and its modelling language is powerful and flexible. Forté Software was acquired by Sun Microsystems and subsequently became a part of iPlanet E-Commerce Solutions in October 1999. In late 2000, version 3.0 of the product became an integral part of iPlanet Integration Server. The evaluation here is based on version 1.0 of the product. The sample process introduced in Figure 3.10, implemented with Forté Conductor, is shown in Figure 3.12.

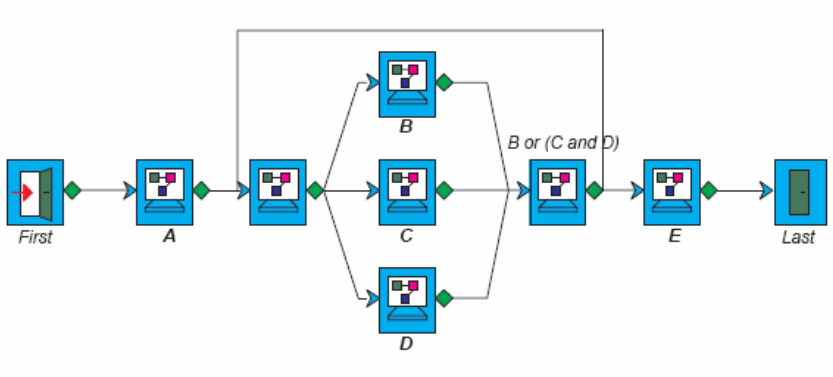


Figure3.12 - Sample process model implemented with Forté Conductor

Changengine

Changengine [HP00] is a workflow offering from HP, the second largest computer supplier in the world. The first major version of the product, 3.0, has been introduced in 1998 and it is focused on high performance and support for dynamic modifications. In late 2000, the product changed its name to HP Process Manager to better convey the purpose of the product to the customers. The evaluation here is based on version 4.0, introduced in early 2000. A business process in Changengine consists of a number of different process elements called Nodes linked by Arcs.
The sample process introduced in Figure 3.10 implemented with Changengine is shown in Figure 3.13.

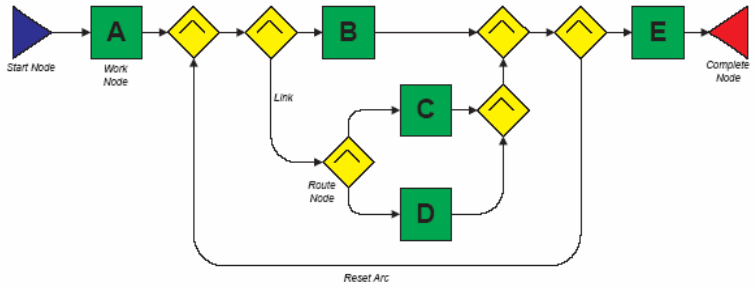


Figure3.13 - Sample process model implemented with Changengine

Staffware

Staffware [Sta00] is one of the leading workflow management systems. Staffware is authored and distributed by Staffware PLC. Staffware PLC has its headquarters in Maidenhead (UK), operates through offices in 15 countries and has a network of 360 partners, resellers and OEMs. There is used the most recent version of Staffware (i.e. Staffware 2000), which was released in the last quarter of 1999. Figure 3.14 shows the sample process introduced in Figure 3.10 implemented with Staffware.

Fujitsu i-Flow

Fujitsu i-Flow [Fuj99] is a workflow offering from Fujitsu that can be seen as the successor of the well-established workflow engine from the same company, TeamWare. I-Flow is web-centric and has a Java/CORBA based engine built specifically for Independent Software Vendors and System Integrators. The evaluation here is based on version 3.5 of the product, introduced in early 2000. The latest version of the product is 4.1 (as of early 2002).
The sample process introduced in Figure 3.10, implemented with i-Flow is shown in Figure 3.15.

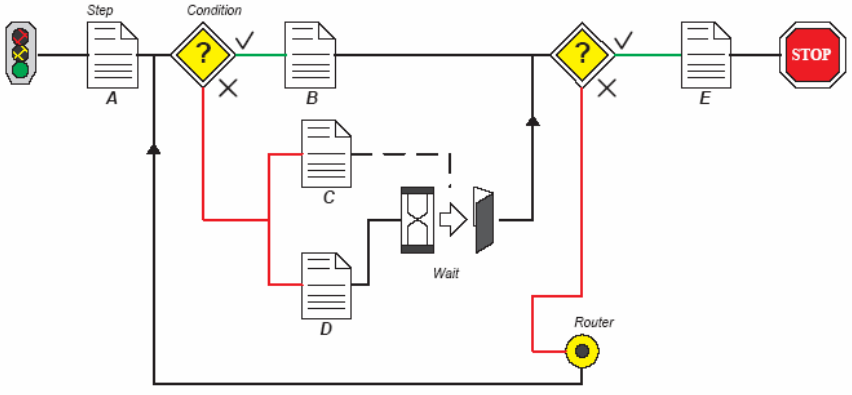


Figure3.14 - Sample process model implemented with Staffware

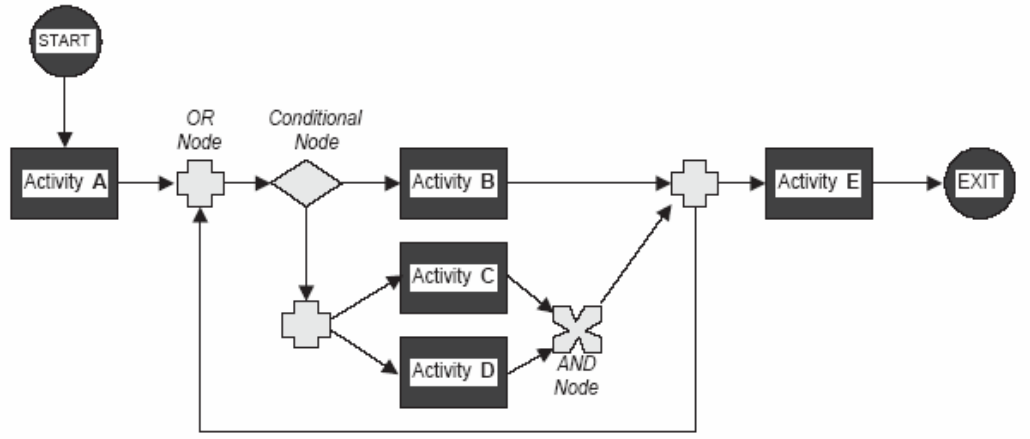


Figure3.15 - Sample process model implemented with i-Flow

MQSeries Workflow

MQSeries Workflow [IBM99] is the successor of IBM's major workflow offering, FlowMark. FlowMark was one of the first workflow products that were independent from document management and imaging services. It has been renamed to MQSeries Workflow after a move from the proprietary middleware to middleware based on the MQSeries product. The evaluation here is based on version 3.1.
Figure 3.16 shows the sample process introduced in Figure 3.10 implemented as an MQSeries Workflow process model.

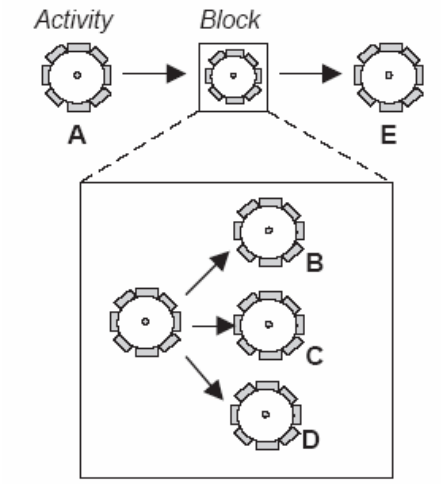


Figure3.16 - Sample process model implemented with MQSeries Workflow

Verve

Verve [Ver00] is a relative newcomer to the workflow market as it made its debut in 1998. In late 2000, it was acquired by Versata and renamed Versata Integration Server (VIS). The evaluation here is based on version 2.0 of the product. What makes it an interesting workflow product is that it has been designed from the ground up as an embeddable workflow engine (i.e. a workflow engine that can be embedded in other applications rather than used as a stand-alone product).
The sample process introduced in Figure 3.10, implemented with Verve is shown in figure 3.17.

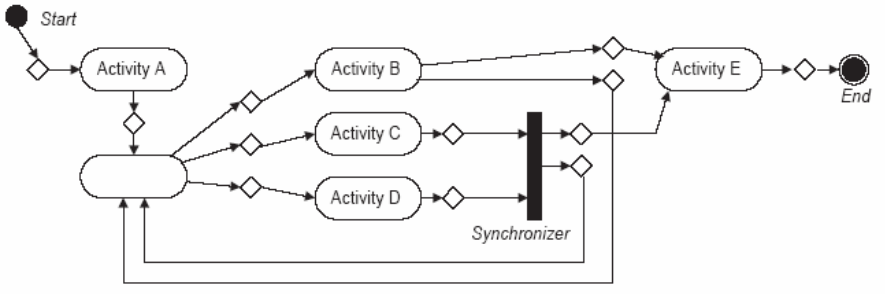


Figure3.17 - Sample process model implemented with Verve

SAP R/3 Workflow

SAP is the main player in the market of ERP systems. Its R/3 software suite includes an integrated workflow component called SAP R/3 Workflow [SAP97] that have been evaluated independently of the rest of R/3. The evaluation here is based on release 4.6 of the product, which is a current release as of the beginning of 2002.
The sample process introduced in Figure 3.10, implemented with SAP R/3 Workflow is shown in Figure 3.18.

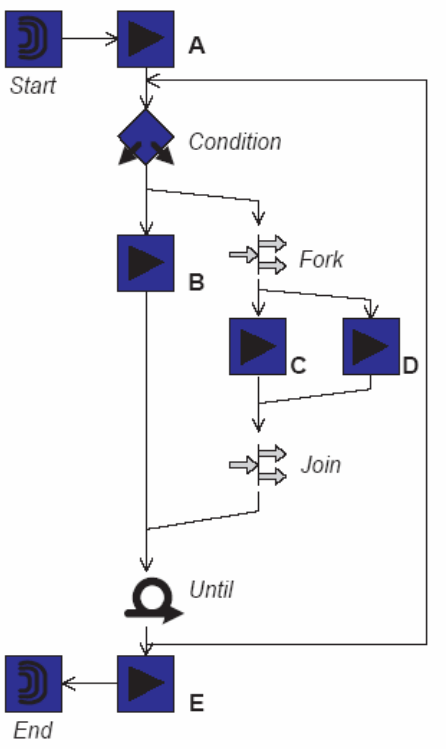


Figure3.18 - Sample process model implemented with SAP R/3 Workflow

The presentations in 3.2.1 and 3.2.2 give examples about the current methodology and the current experience for the evaluation of workflow products and their qualities. The evaluations are performed related to the functional patterns, which are supported by the products. Graphical representations of common workflow scheme are trying to give knowledge about the quality of the software tools. The products are presented in a descriptive way. From user's point of view it is not evident the advantages and disadvantages of the evaluated tools. The criteria, which are used, have academic nature related to the workflow patterns. Thus the recommendations made by such evaluation, address design issue. They do not address explicitly quantitative characteristics, which can help the common user for making relative choice about the application quality of the software tools.

3. 3 Software Evaluation Methodology

This chapter presents an overview of different methodologies, applied for the evaluation of the software products. The overview demonstrates that there are several approaches and

application methodologies for assessment of software products (3.3.1 – 3.3.5). Particular attention is done for those methodologies, which address information systems, and products, which commonly operate in global network environment and Internet surroundings.
The experience gathered in the domain of developing methodologies for assessment of software and informational products is presented by short description of several approaches, applied for software evaluation.

3.3.1. Smith's criteria for evaluation of Internet based software and information resources

The developments in [Smith, 1997] address the problem of the definition of a methodology for the evaluation and comparison of Internet based software products, which implement appropriate information system, offering software and informational resources. The evaluation is performed by applying a set of criteria for Internet based information resources [3.26]. Because the project VISP targets developments and implementation of informational functionalities for Internet based systems, the evaluation scheme has to consider the peculiarities of the informational systems. The basis of the applied evaluation methodology [Smith, 1997] is described below. As a descriptive list of requirements, important features of the products are summarized.

Scope of the product

What items are included in the product? What subject area, time period, formats or types are covered? Is the scope stated through meta information such as an introduction, or only implied? Does the actual scope of the product match expectations? Aspects of the scope include:
Breadth: Are all aspects of the subject covered?
Depth: To what level of detail in the subject does the resource go?
Time: Is the information in the resource limited to certain time periods?
Format: Are certain kinds of Internet resources (for example PDF, FTP) excluded?

Content

Is the information for the product is factual, or opinion?
Is the product an integral, or is it integrated from several other resources?
Specific aspects related to the product include the accuracy, authority, currency and uniqueness of it.
Accuracy: Is the information for the resource accurate? The Internet has become a prime marketing and advertising tool, and it is advisable to ask "what motivation does the authors have for placing this information on the Net". Frequently the answer is that the information is placed to advertise, or support a particular point of view.
Authority: Does the product have some reputable organization or expert behind it? Does the producer have standing in the field? Are sources of information stated? Is the information verifiable? Can the producer be contacted for clarification or to be informed of new information?
Currency: How frequently is the product is updated, or is it a static product? Are dates of update stated, and do these correspond to the information in the product? Does the organization hosting the product appear to have a commitment to ongoing maintenance and stability of the product?

Uniqueness: Is the information for this product available in other forms (for example other sites, WWW, print, CD-ROM)? What advantages does this particular product have? If the product information is derived from another format, for example print, does it have all the features of the original? Have extra features been added? Does it complement another resource, for instance by providing updates to a print source?
Links made to other resources: If the value of the product lies in its links to other resources, are the links kept up to date, and made to appropriate products? Are the links made in such a way that it is clear that an external resource is being referred to.
Quality of writing: Is the description of the product is well written? The quality of writing is important for the content of the product description to be communicated clearly.

Graphic and multimedia design

Is the product is interesting to look at? Do the visual effects and working examples enhance the product, or substitute the product? If audio, video, virtual reality modelling, etc. are used, are they appropriate to the purpose of the product presentation?

Purpose

What is the purpose of the product? Is this clearly stated? Does the resource fulfill the stated purpose?
Audience: Who are the intended users of this product? At what level is the product pitched: a subject expert, a school student? Will the product satisfy the needs of the intended users? Does this user group correspond to the intended audience?

Reviews

What do other reviewing services say about the product? The use of reviewing journals has been a mainstay of collection development in print collections. In the Internet environment the evaluator will need to become familiar with the strengths and weaknesses of the range of virtual sources reviewing the product.

Workability

Is the product convenient and effective to use? An issue in providing access to electronic help documents is whether a help library should just provide links to the originating site, or "acquire" the publication for local access. Poor workability may indicate that the help library should store the data locally, if intellectual property considerations allow this. Aspects of workability include also:
User friendliness: Are any special commands clear? Is help information available? Have user interface issues been addressed, such as menu design, readability of screens, etc.
Required computing environment: Can the product be used with standard equipment and software, or are there special software, password, or network requirements? Has the product been designed to work well with one software and user interface? Images and other multimedia may create problems if users have not installed the correct support.

Searching: How effectively can information be retrieved for the product functionality? Is the product organized in a logical manner to facilitate the functional utilization? Is a useful search engine provided? What operators and ranking features are available? Is the search engine interface intuitive? Does the search engine index the whole product?

Browsability and organization: Is the product organized in a logical manner to facilitate the location of the different functions?

Interactivity: Where interactive features such as forms, menus are provided, do these work? Do they add value to the product?

Connectivity: Can the product be accessed with standard equipment and software, or are there special software, password, or network requirements? Can the product be accessed reliably in network environment.

Cost

Currently open code products are perceived as being "free". However costs do exist, for the market available products.

3.3.2. Belyk's and Feist's Software Evaluation Criteria

Another evaluation scheme for software products, related to wide network exploitation is developed in [Belyk and Feist, 2002]. The characteristics of this evaluation are commented below.

Product Selection Criteria

A series of categories and criteria is suggested for the evaluation of online collaborative tools in the development, delivery, and administration online of software products [ASTD, 2001].

Cost (institutional and user):

- System requirements for the product operation: open platform, platform-specific, server purchased vs. hosted.
- Bandwidth for network exploitation (modem, cable, ADSL, T-1, etc.).
- License fees (scaled per user).
- User software/hardware requirements.
- Peculiarity of the installation of the product

Complexity (user focus):

- Technical support (user manual; frequently asked questions; online and off-line help);
- Collaborative tools if applicable (asynchronous – email, conferencing; synchronous – chat, audio- conferencing, whiteboard, virtual networking);
- Usability (seamless technology; degree of intuitiveness; ease of use; navigation; consistency; stability; functionality)

Control:

- Secured access (password protection; encryption; firewall).
- Personalization .
- Privacy.

Clarity: resolution, size, layout, etc.
Common Technical Framework: interoperability; integration, protocols, standards supported; scalability; platform; file-sharing.

Features: administrator tools (registration; report generation)

Weighing Product Selection Factors

Before selecting specific products, institutions should consider each of the above factors, balancing as far as possible the merits of specific products against the general features of the system. The selection of a specific product requires attention to:

- Software reliability.
- Availability of technical support by the institution to the users.
- Availability of support by the software supplier to the institution and the users.
- Cost to the institution – e.g., is full ownership (i.e., local server support) available for the software?

Matrix weighted evaluation of the product

A quantification approach, using evaluation matrix is applied for the software product evaluation in [3.27]. The evaluation responses may be weighted using points scoring criteria and scorecards. Results can then be compared quantitatively according to the evaluation matrix. The review and the analysis of the responses are recommended to be performed in the following sequence.

- Analyze each evaluation response using a 'score card'.
- Review each requirement listed in the score card and check the answer(s). It is recommended to use a simple 'Yes or No' marking, or a combined weighting and scoring method to indicate to what degree the scorecard requirements are met by the evaluator.
- Repeat the process, using a new scorecard for each software product.

An example of the weighted score card is presented here, Table 3.4.

Table 3.4 Matrix wighted evaluation of the software product

SCORECARD	**Reviewer: Requirements**	**Date: 8/1/06**	**Products: A**	**Software qualification**
Evaluation Criteria	**Weighting**	**Evaluation Score**	**Points**	**Notes**
Workflow functionality	3 x	3	9	good
Purchase order	3 x	3	9	strong
cost	3 x	3	9	excellent
Needs for training	3 x	3	9	limited
TOTAL SCORE				
MAX SCORE				

- The evaluation criteria have to formalize the requirements towards the software products.
- Requirement weighting: Essential (3 x), Desirable (2 x), Nice to have (1 x).

- Evaluation score is based upon reviewing and analyzing the score card. The evaluation marks are: 0 = does not meet requirements; 1 = partially meets requirements; 2 = meets requirements; 3 = exceed requirements.
- The total points for the evaluation are the multiplication: weighting x evaluation score.

It may be obvious from the evaluator scorecard results which software product should be short listed. The preparation of a common comparison matrix, consisting the results of each of the scorecards above, can be a good choice for presentation of the total result in a table form.

3.3.3. Five Criteria For Evaluating Web Pages

The development methodology in [Kapoun, 1998] is directly addressed for the evaluation of Web pages and Web presence**.** Because the domain of operation of the software products of workflow execution address the global network, from usability consideration it is beneficial to cope requirements from Web presence to workflow execution products. The criteria for evaluation of Web pages are discussed in [3.28]. The evaluation is advised to be performed according to table 5.2 with requirements and appropriate recommendations.

Table 3.5 Five criteria for evaluating Web pages

Evaluation of Web documents	**How to interpret the basics**
1. Accuracy of Web Documents • Who wrote the page and can you contact him or her? • What is the purpose of the document and why was it produced? • Is this person qualified to write this document?	Accuracy • Make sure author provides e-mail or a contact address/phone number. • Know the distinction between author and Webmaster.
2. Authority of Web Documents • Who published the document and is it separate from the "Webmaster?" • Check the domain of the document, what institution publishes this document? • Does the publisher list his or her qualifications?	Authority • What credentials are listed for the authors)? • Where is the document published? Check URL domain.
3. Objectivity of Web Documents • What goals/objectives does this page meet? • How detailed is the information?	Objectivity • Determine if page is a mask for advertising; if so information might be biased. • View any Web page as you would an infommercial on television. Ask yourself

• What opinions (if any) are expressed by the author?	why was this written and for whom?
4. Currency of Web Documents • When was it produced? • When was it updated' • How up-to-date are the links (if any)?	Currency • How many dead links are on the page? • Are the links current or updated regularly? • Is the information on the page outdated?
5. Coverage of the Web Documents • Are the links (if any) evaluated and do they complement the documents' theme? • Is it all images or a balance of text and images? • Is the information presented cited correctly?	Coverage • If page requires special software to view the information, how much are you missing if you don't have the software? • Is it free or is there a fee, to obtain the information? • Is there an option for text only, or frames, or a suggested browser

3.3.4. WWW CyberGuide Ratings

An additional approach for the evaluation of the content of Internet based informational resources is described in [3.29]. The content evaluation and the quality of a Web site design is assessed by [Karen McLachlan, 2002a] for instructional purposes and the methodology is available from [3.30].

WWW Cyberguide rating for content evaluation [Karen McLachlan, 2002a]

First look:

- User is able to quickly determine the basic content of the site.
- User is able to determine the intended audience of the site.

Information Providers:

- The author(s) of the material on the site is clearly identified.
- Information about the author(s) is available.
- According to the info given, author(s) appears qualified on this topic.
- The sponsor of the site is clearly identified.
- A contact person or address is available so the user can ask questions or verify information.

Information Currency:

- Latest revision date is provided. Date last revised.
- Latest revision date is appropriate to material.
- Content is updated frequently.
- Links to other sites are current and working properly.

Information Quality

- The purpose of this site is clear: business/commercial – entertainment – informational - news - personal page – persuasion.
- The content achieves this intended purpose effectively. [Karen McLachlan, 2002a
- The content appears to be complete (no "under construction" signs, for example).
- The content of this site is well organized.
- The information in this site is easy to understand.
- This site offers a sufficient information related to my needs/purposes.
- The content is free of bias, or the bias can be easily detected.
- This site provides interactivity that increases its value.
- The information appears to be accurate based on user's previous knowledge of subject.
- The information is consistent with similar information in other sources.
- Grammar and spelling are correct.

Further Information

- There are links to other sites that are related to the user needs/purposes.
- The content of linked sites is worthwhile and appropriate to the user needs/purposes.

WWW Cyberguide rating for Web site design [Karen McLachlan, 2002b]

Speed: the homepage downloads efficiently.

Home page:

- The homepage is attractive, has strong eye appeal.
- The user can tell where he is immediately (clear title, description, image captions, etc.)
- There is an index, table of contents, or other clear indicator of the contents of the site.
- Site sponsor/provider is clearly identified.
- Information/method for contacting sponsor/provider is readily available.
- Copyright date or date site was established is easy to determine.

Ease of navigation:

- User is able to move around within the site with ease.
- Directions for using the site are provided if necessary.
- Directions are clear and easy to follow.
- The links to other pages within the site are helpful and appropriate.
- Internal and external links are working properly (no dead ends, no incorrect links, etc.)

Use of multimedia:

- Each graphic, audio file, video file, etc., serves a clear purpose.
- The graphics, animations, sounds clips, etc., make a significant contribution to the site.

Browser compatibility: site is equally effective with a variety of browsers such as Netscape and Internet Explorer.

Content Presentation:

- There is sufficient information to make the site worth visiting.
- The information is clearly labeled and organized.
- The same basic format is used consistently throughout site.
- Information is easy to find (no more than three clicks, for example).
- Lists of links are well organized and easy to use.

Currency:

- The date of last revision is clearly labeled. Date last revised.
- Out-dated material has been removed.

Availability of further information:

- A working link is provided to a contact person or address for further information.
- Links to other useful Web sites are provided.

3.3.5. General Software Evaluation Criteria

General requirements for evaluation of software without respect of the software nature has been summarized in EC founded project FRETRIS [FRETRIS]. The evaluation criteria have been classified as follows.

General Considerations

Hereafter are described the pointers for evaluating software and also as a source of specific information.

Content & Evaluation

- Who is the audience?
- What is the purpose of the Software product & what does it contain?
- How complete and accurate are the information and the links provided?
- What is the relative value of the Software product in comparison to the range of information resources available for this topic?
- What other resources (print & non-print) are available in this area?
- What are the date(s) of coverage of the site and site-specific documents?
- How comprehensive is this Software product?
- What are the link selection criteria if any?
- Are the links relevant and appropriate for the Software product?
- Is the Software product inward-focused, pointing outward, or both?
- Is there an appropriate balance between inward pointing links ("inlinks") & outward - pointing links ("outlinks")?
- Are the links comprehensive or do they just provide a sampler?
- What do the links offer that is not easily available in other sources?
- Are the links evaluated in any way?
- Is multimedia appropriately incorporated?
- How valuable is the information provided in the Software product (intrinsic value)?

Source & Date

- Who is the author or producer?

- What is the authority or expertise of the individual or group that created this Software product?
- How knowledgeable is the individual or group on the subject matter of the Software product?
- Is the Software product sponsored by an individual or group that has created other Software products?
- Is any sort of bias evident?
- When was the Software product produced?
- When was the Software product mounted?
- When was the Software product last revised?
- How up to date are the links?
- How reliable are the links; are there blind links, or references to Software products which have moved?
- Is contact information for the author or producer included?

Structure

- Does the window follow good graphic design principles?
- Do the graphics and art serve a function or are they decorative?
- Do the icons clearly represent what is intended?
- Does the text follow basic rules of grammar, spelling and literary composition?
- Is there an element of creativity, and does it add to or detract from the window itself?
- Can the text stand alone for use in line-mode (text only) Web browsers as well as multimedia browsers, or is there an option for line-mode browsers?
- Are links provided to Web "subject trees" or directories; lists of subject-arranged Web sources?

Others

- Is appropriate interactivity available?
- When it is necessary; to send confidential information out over the Internet, is encryption (i.e., a secure coding system) available? How secure is it?
- Are there links to search engines or is a search engine attached to (embedded in) the Software product?
- All the question above can have an answer which positively reflect the features and capabilities of Workflow software.
- This is a "toolbox" of criteria that enable Workflow software to be evaluated for use in current commercial activity of transport field actors.

Another criteria are:
Scope: What items are included in Workflow software ? What area, time period, types of information are covered? Does the actual scope of the resource match expectations?:

Breadth: Are all aspects of Workflow software documentation covered?

Depth: To what level of detail of information provided does Workflow software go?

Time: Is the information in the resource limited to certain time periods?

Format: Are certain kinds of Internet resources (for example FTP) excluded?

Content: Is the resource an integral resource updated by the information original source, or has it been abstracted from another source by an administrator, perhaps losing meaning or links in the process? Specific aspects related to the content include the accuracy, authority, currency and uniqueness of a resource.

Accuracy: Is the information in the resource accurate? The information is placed to advertise, or support a particular point of view.

Authority: Does the resource have some reputable organization or expert behind it? Does the author have standing in the field? Are sources of information stated? Is the information verifiable? Can the author be contacted for clarification or to be informed of new information?

Uniqueness: Is the information in this resource available in other forms (for example other sites, print, CD-ROM)? What advantages does this particular resource have? If the resource is derived from another format, for example print, does it have all the features of the original? Have extra features been added? Does it complement another resource, for instance by providing updates to a print source?

Links made to other resources: If the value of the site lies in its links to other resources, are the links kept up to date, and made to appropriate resources? Are the links made in such a way that it is clear that an external site is being referred to. There are potential copyright issues with sites that, for instance, enclose an external link in frames so that the source of the information is unclear.

Quality of writing: Is the text well written? While hypertext linking and multimedia are important elements of the Web, the bulk of the information content on the Web still lies in text, and quality of writing is important for the content to be communicated clearly.

Graphic and multimedia design: Is the resource interesting to look at? Do the visual effects enhance the resource, distract from the content, or substitute for content? If audio, video, virtual reality modelling, etc. are used, are they appropriate to the purpose of the source? A related criterion to graphic design is navigational design, mentioned below in the context of browsability and organization.
Purpose: What is the purpose of the resource? Is this clearly stated? Does the resource fulfill the stated purpose?

Audience: Who are the intended users of this resource? At what level is the resource pitched: a subject expert, a layperson, or a school student? Will the resource satisfy the needs of the intended users? Does the user group have the connectivity to access the resource? Does your user group correspond to the intended audience?

Workability: Is the resource convenient and effective to use? This is the area where criteria for Workflow software resources differ most from print sources.
Aspects of workability include:

User friendliness: Are any special commands clear? Is help information available? Have user interface issues been addressed, such as menu design, readability of screens, etc.

Required computing environment: Can the resource be accessed with standard equipment and software, or are there special software, password, or network requirements? Has the resource been designed to work well with one software and user interface but is difficult to use with others? It is useful to test resources with a variety of browsers and connections. Some resources may pose problems to users who have not installed appropriate software. Images and other multimedia may create problems if users have not installed the correct viewer.

Searching: How effectively can information be retrieved from the resource? Is the resource organized in a logical manner to facilitate the location of resources? Is the organizational scheme appropriate, for example chronological for an historical source, or geographical for a regional resource? Is a useful search engine provided? What operators and ranking features are available? Is the search engine interface intuitive? Does the search engine index the whole resource?

Browsability and organization: Is the resource organized in a logical manner to facilitate the location of resources? Is the organizational scheme appropriate, for example chronological for an historical source, or geographical for a regional resource?

Interactivity: Where interactive features such as forms, cgi scripts etc. are provided, do these work? Do they add value to the site?

Connectivity: Can the resource be accessed with standard equipment and software, or are there special software, password, or network requirements? Can the resource be accessed reliably, or is it frequently overloaded or off-line? Is a local mirror site available, or do international traffic charges have to be incurred?

Cost: Currently the costs of Internet information resources become important. Costs can be divided into
(a) costs of connecting to the resource and
(b) costs associated with the use of the intellectual property contained in the resource.
In terms of (a), users paying traffic charges are already having to consider the costs of connection, and may want include this in criteria for selection, for instance to text based rather than image intensive sites, if the image content is the same.
Increasingly the user will see sites where (b) is a consideration, and a charge is made for the intellectual content of the site. The Internet has created an expectation and an opportunity to make charged services available to end users. Workflow software Providers have a role in negotiating subscriptions and site licenses for organizational access to charged services. If online transactions are used to pay for information, the security of these transactions at a site may become important. Charged services may be available with limited functionality, or for trial periods, for free. Workflow software Providers will need to decide whether to provide the enhanced or the enhanced or limited version.

Responsibilities of Parties involved

In the evaluation process different parties are involved which play special roles.

- software developer that creates the software product
- testing team performs the test in order to evaluate required software quality

- potential clients operate the software product in order to evaluate if it is acceptable for his needs.

Responsibilities of the developer
- designing and implementing the more useful solutions for obtaining the specified quality for the software product,
- test the software product before to be tested by the test team as a third party in the development process of the software product,
- making all the necessary corrections in order to remove the errors occurred during the testing process.

Responsibilities of the client
- establishing necessary legal rights in the software product including documentation for the purpose of evaluation,
- providing information necessary for identification and description of the product,
- specifying the software characteristics and evaluation level to be considered (in cooperation with the testing team) ,
- providing suitable access to computers and other equipment used for operational use of the software product,
- providing general support to the testing team, including informal training and access to suitable staff for informal discussions etc.,
- ensuring the timely supply of the software product including documentation and other material necessary for the evaluation
- informing the testing team of any factors (e.g. a recently found error) - after the evaluation - that would invalidate the evaluation result.

Responsibilities of the testing team
Responsibilities of the testing team are to fulfill the requirements for working procedures of testing laboratories in order to avoid appeal and review procedures in the context of software evaluation.

The testing team shall establish and maintain a *quality system* appropriate to the evaluation of software according to the procedure described in this evaluator's guide. The quality system must be documented in a quality manual.
Topics of the quality manual include:
- General quality procedures and instructions;
- Quality assurance procedures specific for each evaluation;
- Feedback and corrective actions whenever evaluation discrepancies are detected;
- Procedures for dealing with complaints

Evaluation Level Concept
An evaluation level defines the depth or thoroughness of the evaluation in terms of evaluation techniques to be applied and evaluation results to be achieved, both with reference to evaluation objectives (e.g. safety conditions, security constraints, economic risk, availability conditions, application constraints). As a consequence evaluation at different levels gives different confidence in the quality of the software product. The level can be chosen independently of each characteristic.

3.4. Conclusions

The list of available products, addressing the modelling issues as specification, simulation and execution functionality of the workflow management systems is quite long. The products have different level of maturity and they are presented both as market available products, as an open source software suits. The products try to support modelling standards as common execution formats, recommended by the international standardization organization. In this abundance of products and software tools the evaluation and choice of appropriate product is not an easy task. Attempts to make appropriate evaluations have been done. Several of the results are given in this chapter. Unfortunately all these products endure continuous elaboration. New futures are included in the software products. Taking into account that the standardization efforts of the methodological background for the workflow management systems is far from completion, an absolute evaluation and classification is a very ambitious work.

As results from this survey of the workflow products, the consortium developments have been restricted to several prospective representatives of the workflow software management products. For these representatives a thorough examination has been performed. The appropriate works and results are presented in the next Chapter 4.

4. Survey of Software Products, Supporting Workflow Management

This chapter presents the results of the software analysis and trials for a prospective list of software products. It contains description of the tools that have been evaluated in Task 2.2 of the VISP project.

In Chapter 2, it has been performed a survey, identifying the sets of products, related to the workflow domain of activities: specification, editing, simulation and execution of business tasks, their integration management and control. The survey performed meets a very big list of products, which quality is not obvious from descriptive presentation. The user can find a lot of redundancy in the descriptions, respectively several products are given with a few documentation and explanations. An important products' feature is what kind of standards, related to the workflow management, the products can respect and obey. The standardization approach is a vital prerequisite for the integration of distributed and cooperating info services, which is targeted by the developments of the VISP project. Thus, the tools, which have to be implemented in the current development of the VISP project, have to satisfy appropriate standards and requirements. Following this conclusion, an overview of the products, concerning their support to workflow standards is performed. Thus, an extensive list of products is defined, which was used for making a short list with choice for prospective list of products, which are applicable for the VISP project. To make a decision for the product, which would be applied in the VISP project, additional investigations have been done by testing and validating the functional potential of the workflow software products. Because it was unreasonable and unrealistic to perform tests, installations, training, and validation with all available and overviewed products, presented in the previous chapter, the partner consortium decided to prepare a prospective short list of products. This list will be targeted for precise evaluation and comparison. The role of the short list of the software products is to restrict the amount of additional investigations by means to find appropriate products for the project developments.

The short list has been extracted from the set of products, which meet workflow standardization requirements. In Table 4.1 a classification of the products, related to workflow standardization initiatives has been done.

Table 4. 1 - Long list of products, supporting standardization initiatives

products	***standard***			
	XPDL	**BPMN**	**BPEL**	**Others**
@enterprise Web Workflow System				Proprietary
Abydos Suite				Proprietary
ActionWorks Process Builder				Proprietary
ActionWorks Process Engine				Proprietary
ActiveBPEL			+	Proprietary
ActiveWebflow Server			+	Proprietary
ActiveWebflow Professional Designer			+	Proprietary
ADONIS			+	
Agentflow	+			

Amazonas Worflow	+			
Antflow				Proprietary
Apache Agila			+	
Aspen Grove				Proprietary
Aspose Workflow	+			
Axonet				Proprietary
BEA WebLogic product			+	WSCI
Bexee			+	
Bigbross Bossa				Proprietary
BizFlow				Proprietary
Biztalk Server 2006			+	XLANG Proprietary
Bonita	+			Proprietary
Borland Together Designer 2006		+		UML Proprietary
Bossa				Proprietary
Brightwater Software				Proprietary
Breeze				Proprietary
Business Integration Engine				Proprietary
Business Process Management (by COSA)				Proprietary
Cape Clear Orchestrator			+	Proprietary
Captaris Workflow				Proprietary
Carnot	+			Proprietary
CaslonSoft				Proprietary
Codehaus Werkflow				Proprietary
Collaxa Orchestration Server			+	
COLOSA - ProcessMaker				Proprietary
Compuware UnifaceFlow BPM				Proprietary
con:cern				Proprietary
CuteFlow				Proprietary
Dalma				Proprietary
DralaSoft				Proprietary
DynaFlow Inc.		+		Proprietary
Enhydra JaWE	+			Proprietary
Enhydra Shark	+			Proprietary
eTouch Workflow				Proprietary
EventStudio				Proprietary
Ezenia! Inc.				Proprietary
FileNet				Proprietary
FlowRunner				Proprietary
FloSuite BPM Software				Proprietary
Flotiva				Proprietary
Flux				Proprietary
Freefluo				WSFL Proprietary
Fuego BPM	+	+	+	Proprietary
Galaxia Tikiwiki				Proprietary

GridNexus				Proprietary
iApprove by Integrify				Proprietary
iBEST, LLC.				Proprietary
IBM BPWS4J			+	Proprietary
iGrafx BPEL			+	
iLaunch				Proprietary
i-man integrated management				Proprietary
Imixs				Proprietary
Incodea Corporation - Business Pilot™.				Proprietary
infoRouter				Proprietary
iKE				Proprietary
Ils/process				Proprietary
Interstage BPM	+	+	+	Proprietary
IvyTeam – Xpert.ivy				Proprietary
ivyGrid				Proprietary
JBoss jBpm				UML Proprietary
JFolder				Proprietary
JOpera			+	
K2.net Enterprise Workflow				Proprietary
Kepler				Proprietary
Kintana				Proprietary
Legistar				Proprietary
Lexmark Document Distributor				Proprietary
Lotus Domino Workflow				Proprietary
Metastorm				Proprietary
Micro-Workflow				Proprietary
MidOffice			+	
MQSeries Workflow				Proprietary
MyControl Workflow Server				Proprietary
Newgen	+			Proprietary
Nsite - Workflow On Demand				Proprietary
Oak Grove Systems, Inc. - LegaSuite BPM				Proprietary
OFBiz Workflow Engine	+			
Open Business Engine	+			
OpenFlow				Proprietary
OpenLink - Virtuoso Universal Server				Proprietary
OpenSymphony - OSWorkflow				Proprietary
OpenWFE				Proprietary
Oracle BPEL Process Manager			+	Proprietary
Orbis Software Ltd - Orbis TaskCentre				Proprietary
Pallas Athena - FLOWer				Proprietary
Parasoft BPEL Maestro			+	
Pi Calculus for SOA				Proprietary
PXE			+	
PL/FLOW workflow engine				Proprietary
Popkin Software - BPM				UML

Process Orch. Platform - Portellus BRMS				Proprietary
RoleFlow - 1. Process Designer; 2.Process Manager ; 3 .Monitoring and Reporting				Proprietary
SAP NetWeaver Exchange Infrastructure			+	Proprietary
Savvion - Savvion BPM Studio				Proprietary
Singularity Ltd - Singularity Process Platform				Proprietary
Skelta Workflow.NET				Proprietary
SPA (Scientific Process Automation)				Proprietary
Sparta Systems - TrackWise				Proprietary
Stellent Imaging and BPM				Proprietary
Sun WSCI Editor				WSCI
Syrup				Proprietary
Taverna				WSFL Proprietary
TIBCO - Staffware Process Suite				Proprietary
TQS Europe - ActiveModeler				Proprietary
Triana				Proprietary
Twister			+	
Ultimus - BPM Suite				Proprietary
Versata				Proprietary
Vignette Process Workflow Modeler	+			
W4 Global - W4 Suite				Proprietary
Web and Flo Kontinuum				Proprietary
WebAsyst Issue Tracking				Proprietary
webMethods Business Process Management				Proprietary
WebSphere MQ Workflow			+	Proprietary
Werkflow				Proprietary
wfmOpen	+			Proprietary
wftk (workflow toolkit)				Proprietary
WinOcular - WinOcular Workflow				Proprietary
Workflow Systems - RAPID				Proprietary
WorkMovr: A-Frame				Proprietary
WorkPoint				Proprietary
Workflow Editor - WorkTicket.com				Proprietary
XFlow2				Proprietary
YAWL				Proprietary
ZBuilder				Proprietary
Zebra				Proprietary

As a result from the survey, performed in Chapter 2 it is common wise to select several representatives of the available set of products and to continue their evaluation including installation, configuration, and trial test. The short list of the candidate products has been chosen according to Table 4.2. The reason for this choice has been based on:

- available current experience of the partners, involved;
- the recommendations in the web for different open source products;
- the availability of market and commercial workflow tool, allowed for test and trial purposes

The testing and the evaluation activities of the software products have been allocated as follows:

Table 4. 2 - Workflow products, allocated per partners for test and assessment

Partner	Valtech	UAM	FhI Fokus	Metaware	WUT	ICCS
Software	• jBpm • Biztalk Server • SAP NetWeaver Exchange Infrastructure • Cape Clear	• OpenWFE • Enterprise Xtention	• Oracle BPEL Process Manager • Borland Together Designer 2006 • Process Modeler 2.1 for Microsoft VisioT	• Enhydra Shark • WfMOpen workflow engine • Freefluo • ActiveBPEL Engine	• YAWL • Con-cern	• Active Endpoints ActiveWebflow Standard • ActiveWebflow Designer • Enhydra JaWE • Enhydra Shark • ObjectWeb BONITA

The software products, given in Table 4.2 have been rearranged in an alphabetical order according to Table 4.3.

Table 4. 3 - Alphabetic order of the short list products, evaluated by the VISP consortium

1	ActiveBPEL Engine (Metaware)
2	Active Endpoints ActiveWebflow Standard (ICCS)
3	ActiveWebflow Designer (ICCS)
4	Biztalk Server (VALTECH)
5	Borland Together Designer 2006 (FhI Fokus)
6	Cape Clear 6.5 (VALTECH)
7	con-cern (WUT)
8	Enhydra JaWE (ICCS)
9	Enhydra Shark (ICCS, Metaware)
10	EnterpriseXtention (UAM)
11	Freefluo (Metaware)
12	JBoss jBPM (VALTECH)
13	ObjectWeb BONITA (ICCS)
14	OpenWFE (UAM)
15	Oracle BPEL Process Manager (FhI Fokus)
16	SAP NetWeaver Exchange Infrastructure (VALTECH)
17	WebAsyst Workflow Management Software (ICCS)
18	WfMOpen (Metaware)
19	YAWL (WUT)

4.1. ActiveBPEL Engine

Workflow software overview

Workflow software presentation

ActiveBPEL Engine (used in thousands of organizations worldwide) is an open source workflow engine written in Java, distributed under the GNU General Public License (GPL) and is a robust runtime environment that is capable of executing process definitions created to the Business Process Execution Language for Web Services (BPEL4WS, or just BPEL) 1.1 specifications. The ActiveBPEL engine technology is developed and maintained by Active Endpoints.

Workflow software description

The ActiveBPEL engine executes Business Process Execution Language (BPEL) processes. It accepts BPEL process definitions, creates process instances, and executes them. There are three main areas in the architecture of the ActiveBPEL engine: the *engine*, *processes*, and *activities*. The ActiveBPEL engine coordinates the execution of one or more BPEL processes. Processes are in turn made up of activities, which may in turn contain or link to further activities.

The architecture of engine is depicted below:

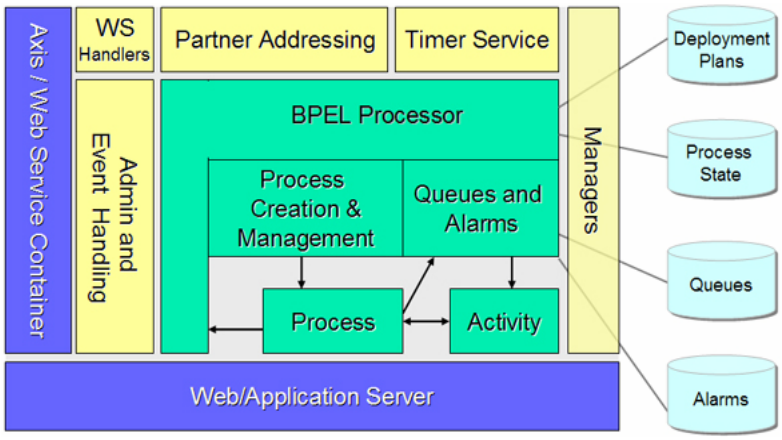


Figure 4.1 - ActiveBPEL engine

Category of the software product

The ActiveBPEL engine is an Open Source implementation of a BPEL engine, written in Java. It reads BPEL process definitions (and other inputs such as WSDL files) and creates representations of BPEL processes. When an incoming message triggers a start activity, the engine creates a new process instance and starts it running. The engine takes care of persistence, queues, alarms, and many other execution details. The ActiveBPEL engine runs in any standard servlet container such as Tomcat 5.5 and supports Java 1.5. Moreover, the engine boasts automation features, such as static analysis, process persistence and process versioning.

Supported interfaces

Interface 1: Exchange of workflow descriptions (models, choreographies, orchestration) between the tool and other modelling tools, repositories, workflow execution tools.
The ActiveBPEL engine can import either 1.X or 2.0 Process Deployment Descriptor (PDD), Partner Definitions (PDEF), WSDL BPEL and BPRD deployment files.

Interface 2: Supports the invocation of workflow engines by workflow clients; Implementation of the interface (by WSDL interface or a programming language interface).
The workflow engine can be invoked using distribution java API or by SOAP calls.

Interface 3: Application invocation, which perform additional workflow activities; Implementation of the interface (by WSDL interface, programming language interface; notification mechanism for human actions).
The engine can invoke Web services, which are associated to relative WSDL. The engine hands Web services to manage both synchronous and asynchronous communication.

Interface 4: Support of workflow integration and collaboration (between different domains and/or distributed workflow execution platforms.
Not applicable.

Interface 5: Monitoring and administration of the workflows activities; Life cycle management of distributed workflows.
The ActiveBPEL Engine can be managed by BpelAdministration tool. It is a web based application, which permits to manage workflow activities, configure engine parameters, persistent storage, deployed processes, partner definitions, processes status, alarm and receive queues.

Additional interfaces: import/export of data towards other products like databases, file systems, repositories (UDDI, LDAP); monitoring of transactions.

Supported standards

Conformity to standards and exchange formats
The engine supports BPEL 1.1 standard and WSDL catalogs. Additionally it supports PDD Format, URL and URN resolutions, Tamino XML database, XPath and XQuery APIs, JavaScript and Apache Bean Scripting Framework (BSF).
The engine version 2.0 supports new BPEL activities (the Break & Continue and Suspend activities) as defined in the new BPEL 2.0 standard.
Support for additional WS-* specifications including:

- WS-Security
- WS-ReliableMessaging
- WS-Addressing
- WS-Policy

4.2. ActiveWebflow™ Standard

ActiveWebflow™ Standard (AWS) is a complete BPEL engine running either on top of a J2EE application server or standalone with a web servlet container (such as Tomcat) [4.1]. AWS servers are high performance BPEL servers that deliver many enterprise features including static analysis, process persistence, process versioning, extensive runtime web console, programmable Web Service and Java APIs plus diagram-based diagnostics.

Key Benefits:

- **Accuracy.** AWS servers support multilevel validation including validity verification of BPEL process definitions and of the incoming message's format.
- **Speed**. AWS servers are fast as they use BPEL as their native process language and do not have to translate between BPEL and proprietary internal constructs.
- **Reliability**. AWS servers support process persistence, which caches state information at critical points during process execution. If a server goes down, either unexpectedly or for maintenance, process executions will resume automatically when the server is returned to operation.
- **Growth.** AWS servers provide comprehensive versioning features. AWS servers use the new version for all new process instances and older versions for in-flight instances. Once all in-flight instances have completed, the older process versions are automatically retired.
- **Control.** When it comes to complex process management applications, there is no one-size-fits-all technology. Recognizing this reality, Active Endpoints has exposed a wealth of internal server APIs via Web services. What's more, it is the first commercial BPEL product company to commit the source code of its core engine technologies into open source through ActiveBPEL initiative, giving the necessary power and control to understand the internal workings of AWS servers.

ActiveWebflow™ Standard - Main features

AWS Server Features

AWS provides an enterprise-level business process execution engine that runs as an application in the Tomcat server. AWS can be used to:

- Execute BPEL4WS 1.1 (BPEL) compliant processes
- Manage the server's configuration
- Monitor and analyze process execution using the server's management consoles
- Suspend, resume or terminate running process instances
- Manage the effective and expiration dates of process versions

In addition the processes can be debugged visually using ActiveWebflow Professional Tools. The main features of **AWS** server can be summarized in the following Table 4.4.

Table 4. 4 - Functional peculiarities of AWS tool

Feature	Description
Fault Tolerance	AWS provides the fault tolerance capabilities that are required to support standard-class process deployments. Using standard database servers, ActiveWebflow persists the information about a process and all running instances that allow for the recovery of a process' state if a server is shut down for any reason. The process information that is persisted includes: Definitions, State, Variables (data), Alarms.
Web Services Support	AWS includes a J2SE compliant mechanism for handling Web service requests and Web service invocations. In addition to ActiveWebflow's built-in support, you can optionally use any Web service container running in Tomcat to deploy the process's services. You can also provide a unique override per Partner Link for invoking services.

Process Versioning	AWS provides an implementation of process versioning that is flexible and robust. The following version parameters can be specified: • Version number -- default is to automatically increment if the process has significantly changed and no version number provided • Effective Date -- default is to take effect immediately if not set • Expiration Date -- default is to not expire • Running process instance disposition • Maintain (default) -- allow running instances to complete using the definition that they started under • Migrate -- migrate all running instances to the new definition (definitions must be compatible) • Terminate -- terminate all running instances

AWS Management Features

AWS provides comprehensive management and administration capabilities. Web-based consoles allow the configuration and administration of the server's execution environment and deployed and running processes. The console pages are grouped into three categories. The Home page provides an overview of the server including the type of configuration, license status and engine running status.

Engine. The Engine console pages allow to be viewed and managed settings of the engine, main of which are listed in the following Table 4.5:

Table 4.5 - Settings of AWS engine

Engine Settings	Description
Configuration of ActiveWebflow Standard	• Enable and disable process instance logging • Turn on and off message validation • Enable or disable BPEL extensions • Set engine caching parameters
Detailed license information	Add and remove licenses
Persistent storage maintenance	• Prune completed processes by date • Prune deployment logs by date • Prune deployed process definitions
Detailed information regarding the version of ActiveWebflow Standard installed	Version and build numbers for all ActiveWebflow Standard components

Engine Status Information. When the engine is running, there are two Web pages available that display information about the server: the Axis Web services listing all the available services, and the ActiveBPEL engine administration page.

Deployment

The Deployment console pages allow the processes to be managed as well as their artifacts that are deployed to the ActiveWebflow Standard server. The main functions are:

- Deployment of Business Process Archive (.bpr) files
- Deployment logs for each .bpr file for errors and warnings.

- Lists of all deployed processes with selection filters
- Detailed version information including the ability to modify specific data based on the version state (e.g., status, effective and expiration dates)
- Detailed information regarding other artifacts deployed with process
- WSDL files deployed in the BPR
- Global Partner definition files deployed to the server

Process Status

The Process Status console pages allow to view and manage information associated with individual process instances. These pages provide the ability to sort and filter the information in ways that make easier to deal with only the most relevant process information. For running and completed processes the designer can use the consoles to perform the following tasks:

- Suspend, Resume and Terminate executing processes
- View the process execution state graphically, hierarchically, and textually
- Examine the content and status of process variables, activities and links

For the persistent queues that allow for the re-hydration of processes on server startup one can inspect the values of:

- Alarms: events that will execute at a specific time as specified by executing processes.
- Receives: the result of a BPEL activity or Event Handler specifying that a message is expected.

Supported Interfaces

Interface 1. AWS uses BPEL descriptions for the processes and WSDL descriptions for the services. They files can be created manually or using ActiveWebflow Designer.

Interface 2. ActiveBPEL engine starts with the servlet container.
In order for the engine to execute a BPEL process, several files must be packaged in a business process archive (.bpr) file for deployment. The basic business process archive includes:

- BPEL process (.bpel file)
- WSDL file(s) referenced in the BPEL process (.wsdl file)
- WSDL catalogue (.xml file)
- Process deployment descriptor (.pdd file)
- Optionally, a partner definition file (.pdef file)

These files can be created manually or using ActiveWebflow Designer.

Interface 3. WS applications with a valid partner link type in the associated WSDL can be invoked using generic SOAP client and Java client application.

Interface 4. Support of workflow integration and collaboration - The collaboration should be possible and be based on SOAP/WSDL standards.

Interface 5. AWS provides comprehensive management and administration capabilities. Web-based consoles allow the configuration and administration of the server's execution environment and deployed and running processes.

4.3. ActiveWebflow™ Professional Designer

Introduction

ActiveWebflow™ Professional Designer - is a comprehensive process management solution for organizations that need to create and deploy complex services-oriented integrations. Based on the BPEL4WS 1.1 specification and schema, ActiveWebflow includes a high-productivity, Eclipse-based visual design environment and an enterprise-class server [4.2] www.eclipse.org.

Description

ActiveWebflow Designer (**Designer**, for short) is a visual environment for creating, testing and deploying BPEL-based process compositions. The **ActiveWebflow Designer** is a plug-in to the Eclipse [4.3] ActiveWebflow™ Enterprise and Standard Server User's Guide, [4.4] integrated development environment. **ActiveWebflow** takes advantage of the Eclipse Workbench features to provide BPEL building capabilities.

Main features:

- Creating BPEL processes:
  - processes are built by choosing partners, services and operations, and defining how data flows among those entities;
  - to do so icons are organized on the Process Editor canvas;
  - in the same time ActiveWebflow constructs valid BPEL code.
- Testing processes – Simulation and Debugging:
  - Simulate process execution using sample data;
  - Set breakpoints, step through or run the process;
  - Remotely debug a process running on the server.
- Deploying created processes:
  - endpoint references for services used in a process are provided by Deployment wizard;
  - A process deployment descriptor provides error-free techniques for binding the created services.
  - Processes can be automatically deployed to the appropriate server location.

Figure 4.2 presents a typical picture of an **ActiveWebflow** BPEL process.

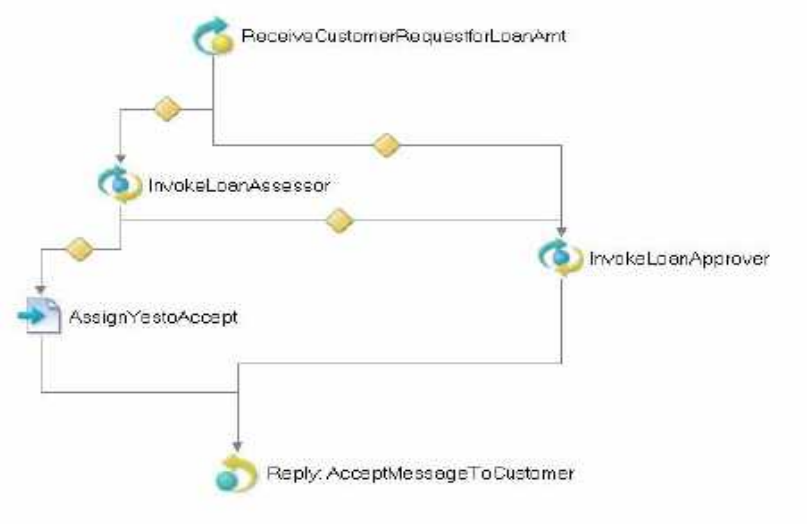


Figure 4.2 - ActiveWebflow BPEL process

The following snippet is the BPEL code corresponding to the illustration above, as generated by **ActiveWebflow** Designer.

```
<?xml version="1.0" encoding="UTF-8"?>
<!--
BPEL Process Definition
Edited using ActiveWebflow(tm) Designer version 1.0.0 (htt
-->
<process name="loanApprovalProcess" suppressJoinFailure="y
   <partnerLinks>
      <partnerLink myRole="customer" name="customer" partn
      <partnerLink name="approver" partnerLinkType="lns:lo
      <partnerLink name="assessor" partnerLinkType="lns:ri
   </partnerLinks>
   <variables>
      <variable messageType="lns:creditInformationMessage"
      <variable messageType="lns:riskAssessmentMessage" na
```

Figure 4. 3 - Generated BPEL code

Functionality

The main functionality of **ActiveWebflow** Designer can be summarized in the following table.

Table 4. 6 - Functionalities of ActiveWebflow Designer

Feature	Description
Simultaneously displayed diagrammatic and hierarchical view of process	Create a process diagrammatically on the canvas. Use the synchronized Outline view to see a hierarchical element structure of the process. Source view is also available to view BPEL 1.1 code that ActiveWebflow generates.
Process Editor Canvas	Drag and drop icons onto a canvas to create a process. ActiveWebflow creates valid BPEL code and generates a task list for missing and invalid activity properties.

Web References	Add the necessary WSDL files to the Web References view for automatic discovery and organization of all pertinent information stored in existing WSDL. Comprehensive searching is available to locate namespaces, messages, and other elements. Drag and drop operations to the Process Editor canvas for automated activity creation.
Management of Sample Data	Add sample data files for all WSDL messages for a convenient repository of test data across all processes using the messages. Add multiple files for various test scenarios. During simulation, test various execution paths using different data.
Automatic Task Generation (static analysis)	ActiveWebflow generates a problem list for all incomplete or invalid BPEL constructs. Thus, the problems can be fixed without hunting for them. This feature works on imported BPEL files as well as native ones.
Create Partner Link Types	Add partner link types to an existing or new WSDL file by using the partner link type wizard. Partner link types are WSDL extensions required for BPEL processes.
Swimlanes	Visual display of each partner's role in the process to show a service is being invoked, received from or replied to. Export information to share with partners.
Automatic Variable Assignment	Create Copy Operations automatically for new or existing Assign activities. Drag the Copy FROM variable to the Copy TO variable. Icons and colors indicate at a glance how a variable is used.
Expression and Query Builders	ActiveWebflow provides visual expression editing controls for building a wide range of scripts. ActiveWebflow's expression editor can be readily extended in order to be included custom functions. Built-in BPEL functions are automatically added to expressions as appropriate.
Activity Properties	Required and optional activity attributes are grouped for easy selection in the Properties view. Pertinent selections are in drop-down lists. Add comments, if desired. Add correlation properties, compensation, and fault handling.
Create BPELets to re-use a selection of activities in other processes	Select one or more activities on the Process Editor canvas and save them to the Custom Palette for later use. Significantly shorten design time by reusing modular elements.
Simulation and Debugging	Simulate process execution using sample data. Set breakpoints, step through or run the process. Remotely debug a process running on the server.
Process Deployment	Deployment wizards guide the user to provide endpoint references for services used in the designed process. A process deployment descriptor provides error-free techniques for binding your services. Process is automatically deployed to the appropriate server location within a package that contains all required files.

In **ActiveWebflow**, one can build two kinds of business processes: Abstract and Executable. The following Figure 4.4 shows that one can start the process either by using WSDL files in existence or by building an abstract BPEL process first and then create the WSDL files for it.

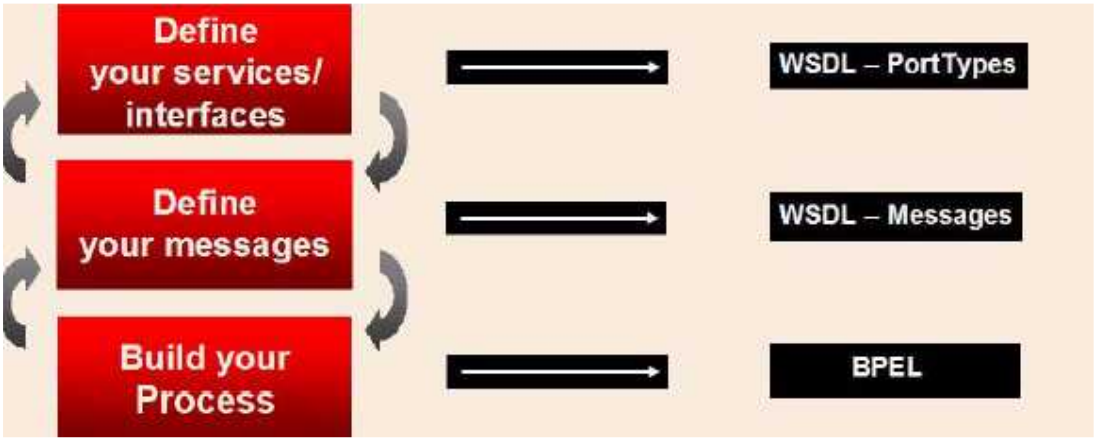


Figure 4.4 - Creating processes

Interfaces – according to Workflow Reference Model [4.5]

Interface1

ActiveWebflow Designer provides interfaces at key process definition points:

- Create new BPEL activities via drag-n-drop from business operations defined in Web Services Description Language (WSDL) files;
- Generate BPEL-specific WSDL extensions for Partner Link Types;
- Generate BPEL-specific WSDL extensions for message properties and property aliases.

Interface2

ActiveWebflow Designer generates a *.bpel* file from the created diagram and generates *.wsdl* file for the diagram itself.

4.4. Biztalk Server

Workflow software overview

BizTalk Server is an Integration Server product used to connect different applications, systems or business processes within an organization or between organizations. It is also used to aggregate services to build a Service Oriented Architecture (SOA).

Category of the software product

BizTalk 2006 provides a full environment for modelling and processing orchestration processes. This environment relies on other Microsoft products, such as Visio for the process modelling.

Supported interfaces

Interface 1: It is possible to import BPEL definitions in BizTalk Server 2006. It provides full support to XLANG (Microsoft business process' definition language).

Interface 2: Clients can be of various kinds: .Net client, any client aware of Web Service (if the business process is deployed as such), mail clients.

Interface 3: BizTalk provides a suite of adapters to integrate various application types: Net applications, SAP, MSMQ, Oracle Applications, PeopleSoft

Interface 4: A process deployed on BizTalk can be part of a collaboration workflow (support for BPEL, XLANG).

Interface 5: BizTalk ships with a Monitoring / Notifier tool (Business Activity Monitor). A portal gives access to process instances statistics, information. One can also define alerts, which can be sent by email for instance. Finally, BAM exposes web services interface to create alerts, retrieve BAM configurations.

Additional interfaces: BizTalk supports import/export to UDDI directories via a wizard.

Supported standards

a. Process definition tools: The business processes are stored as XLANG definitions. BPEL is not natively supported: that means BPEL definitions that are imported into BizTalk are converted to XLANG definitions. Microsoft also gives recommendations to developers to export processes definitions safely (by not using such or such functionality in XLANG).

b. Technical environment is Windows Server platform with .Net framework and SQL server. Figure 4.5 summarizes all the dependencies.

Feature Tree Selection	Platform Dependency: Internet Information Services (IIS)	Windows SharePoint Services 2.0 SP2	Visual Studio 2005	SQL 2000: SQL Server 2000 SP4	SQL Server 2000 Client Tools	SQL Server 2000 Analysis Services SP4	SQL 2005: SQL Server 2005	SQL 2005 Client Tools	SQL Server 2005 Analysis Services	SQL Server Notification Services	Internet Explorer 6.0 SP1	Microsoft Management Console 3.0	Office Excel 2003 SP2	Office InfoPath 2003 SP2	Office Visio 2003 SP2	.NET Framework 1.1	CAB File Components: .NET Framework 2.0	MSXML 3.0 SP7	MSXML 6.0	ADO.NET 8.0	ADO.NET 9.0	SQLXML 3.0 SP3	MDAC 2.8	Office Web Components (OWC) 11
Documentation																	●							
Server Runtime											●						●		●	●	●	●	●	
Base EDI Adapter											●						●							
Administrative Tools											●						●		●	●	●	●	●	●
Developer Tools and SDK			●								●						●		●	●	●		●	
Portal Components																								
Business Activity Monitoring	●				●			●									●							
Business Activity Services	●	●		●			●				●			●			●							
Human Workflow Web Service	●																●							
Additional Software																								
Enterprise Single Sign-On Master Secret Server																	●							
Enterprise Single Sign-On Administration												●					●							
Business Rules Components																	●							
MQSeries Agent																								
Windows SharePoint Services Adapter Web Service																	●							
BAM Alert Provider for SQL Notification Services	●	●								●							●			●	●			
BAM Client				●	●	●	●	●	●				●				●							
BAM Event API																	●							
Other Software																								
Business Rule Engine											●						●							
ODBA															●		●							

Figure4.5. Biztalk Server's supported standards

The following figure shows the main components of the BizTalk Server engine:

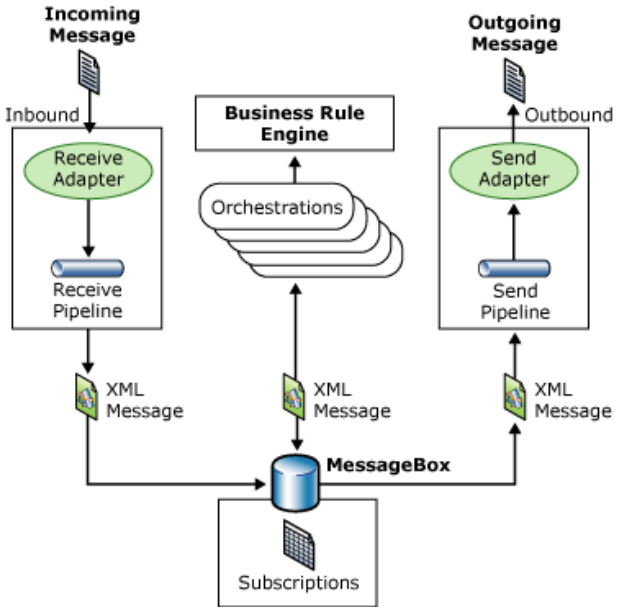


Figure4.6. Components of the BizTalk Server engine

A business process is implemented as one or more orchestrations, each of which consists of executable code. Orchestrations are not created by writing code in a language such as C#. Instead, a business analyst uses the Orchestration Designer for Business Analysts (a Microsoft Visio® snap-in) to organize graphically a defined group of shapes to express the conditions, loops, and other behaviour of the business process. Business processes can also use the Business Rule Engine, which provides a simpler and more easily modified way to express the rules in a business process.

Each orchestration creates subscriptions to indicate the kinds of messages it receives. As illustrated in the preceding figure, the message processing proceeds as follows:

1. A message is received through a receive adapter. Different adapters provide different communication mechanisms, so a message might be acquired by accessing a Web service, reading from a file, or in some other way.
2. The message is processed through a receive pipeline. This pipeline can contain components that perform tasks such as converting the message from its native format into an XML document and validating its digital signature.
3. The message is delivered to a database called MessageBox database, which is implemented by using Microsoft SQL Server.
4. When a message arrives in the MessageBox database, that message is dispatched to its target orchestration, which takes whatever action the business process requires.
5. The result of this processing is typically another message, which is produced by the business process and saved in the MessageBox database.
6. The new message is processed by a send pipeline, which may convert it from the internal XML format used by BizTalk Server 2004 to the format required by its destination, add a digital signature, and more.

7. The message is sent out through a send adapter, which uses an appropriate mechanism to communicate with the application for which this message is destined.

Because applications communicate in many different ways, the BizTalk Server engine supports a variety of protocols and message formats. It is important to remember that the engine works only with XML documents internally. All messages are converted to XML documents when they are received. Similarly, if the recipient of a document requires a format other than XML, the engine converts the document into the appropriate format.

The BizTalk Server engine uses adapters to talk to a wide range of other software. An adapter is an implementation of a communication mechanism, such as a particular protocol. BizTalk Server provides several built-in adapters, and other adapters have been created for popular applications such as SAP. A developer can determine which adapters to use in a given situation, or can create custom adapters.

All adapters are built on a standard base called the Adapter Framework, which is new with BizTalk Server. This framework provides a common way to create and run adapters, and enables you to use the same tools to manage both standard and custom adapters.

4.5. Borland Together Designer 2006

Introduction

Borland Together Designer 2006 is a modelling tool supporting UML, MDA like model transformations utilizing QVT, and business process modelling utilizing the Business Process Modelling Notation (BPMN). It supports the most frequently needed diagrams and notations defined by UML 1.4 like:

- Component diagram
- Use case diagram
- Deployment diagram
- Activity diagram
- Sequence diagram
- Collaboration diagram
- Statechart diagram
- Class diagram includes packages

Together Designer 2006 also provides support for the most frequently notations defined by UML 2.0 like:

- Activity diagram
- Class diagram
- Component diagram
- Composite structure diagram
- Deployment diagram
- State Machine diagram
- Use case diagram
- Interaction (Sequence and Communication) diagrams

In the context of VISP, especially the ability to model business processes has been evaluated. *Together Designer 2006* contains a user-friendly graphical modelling editor supporting

BPMN. The business process diagram tools palette in the editor provides all necessary elements for modelling business activities. The user can drag and drop elements to a BPMN diagram and set after that their properties in a special window. *Together Designer 2006* supports BPMN, with some minor changes related to mapping BPMN to BPEL 1.1.

BPMN is designed to cover many types of business process modelling with any level of detailing and allows the creation of end-to-end business processes. A diagram created in the BPMN project can later be mapped and exported to BPEL/WSDL files with the help of the *Export* command on the *File* menu.

Together Designer 2006 provides a special view for the BMPN validation. After the validation, the tool creates a list of problems with information about potential reasons. From the context menu, the user can quickly find on the diagram the element, which could not pass the validation. The tool provides comprehensive documentation and examples for the business process modelling on BPMN.

The following BPMN elements are supported and can be mapped to BPEL 1.1. Details about BPMN are provided in deliverable D2.1.

Process Structure: Flows and Sequences

- Starting and Ending of Business Processes
- Executing Tasks Sequentially
- Executing Tasks in Parallel
- Flow Inside Flow
- Exclusive Gateways
- Exception Flows (non standard mapping)

Elements

- Business Processes
- Events
- Activities
- Gateways
- Message Flow

The following list shows the BPMN elements that are not supported in the evaluated version:

- Artifacts: The mapping is not provided by specification.
- Complex gateways: The mapping is not provided by specification.
- Lane: The mapping is not required by specification.
- Cancel events: The mapping is not provided by specification.
- Inclusive gateway: It is not supported in this version.
- Manual tasks and script tasks: Their implementation is server dependent.
- Independent subprocesses: Service tasks should be used instead.

Pools and Message Flows

WSDL files are generated for the diagram itself and for all pools, which represent the abstract or private processes. The WSDL is not generated in two cases:

- The pool has attached WSDL

- The pool represents the abstract process, which does not contain any Web services (consumer processes). In other words, a pool, which does not contain any elements (except lanes) and does not have incoming message flows other than reply for outgoing message flows.

Category

Together Designer 2006 is a modelling tool.

Interfaces

As a modelling tool *Together Designer 2006* does not support any of the interfaces identified in the WfMC reference model. Instead, it exports abstract BPEL 1.1 specification that can be refined utilizing a BPEL modelling tool.

4.6. Cape Clear 6.5

Workflow software overview

Cape Clear 6.5 is the latest version of the Product suite dedicated to Service Oriented Architecture (SOA). This company has been founded in 1999 by IONA senior executives and released the latest version in December 2005.

Cape Clear 6.5's Description

Cape Clear 6.5 is comprised of five tightly integrated components, Figure 4.7.

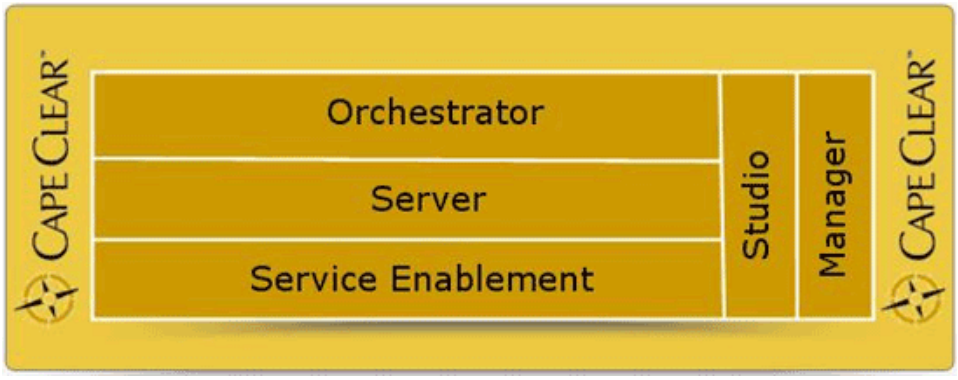


Figure 4.7 - Cape Clear 6.5's Structure

Service Enablement	Innovative tooling to expose critical business functionality and data so that it can be reused in new composite applications.
Server	- Enterprise-class configuration-based service mediation capabilities, such as transformation and routing, to ensure loosely coupled, flexible applications. - Support for multi-protocol messaging and routing. Deploy Cape Clear Server to various applications servers or standalone. - Supports the latest security standards, like WS-Security and SAML
Orchestrator	Framework and tooling for orchestrating services into automated

	business processes : - Cape Clear Orchestrator includes a graphical, Eclipse-based framework for designing and constructing workflows and service orchestrations using BPEL 1.1. Long-running business processes are supported through automatic data persistence and message correlation. - A browser-based console offers detailed management and monitoring of deployed processes. -
Studio	Unified Eclipse-based environment supporting the end-to-end development of SOA-based applications : - Cape Clear Studio incorporates an integrated environment to test your services and processes. In addition, JUnit is bundled with Studio for off-line, automated testing. - Configure and manage servers within Cape Clear Studio. "One-click" package and deploy (and hot re-deploy) of services to the Cape Clear runtime is provided. Remote deployment is also fully supported. - Graphical WSDL and schema editors - Wizards generate Java Web service skeleton classes, to which business logic is added using sophisticated Java Web services tooling.
Manager	Browser-based tool providing configuration, monitoring, diagnostics, and troubleshooting capabilities for your SOA : - Easiness to configure servers and clusters and additional infrastructure, like adapters, transports, and integration with legacy messaging systems. - Manage service life cycle and configuration for all hosted services with Cape Clear Manager. - provides comprehensive insight into running systems and includes support for logging, traps, alerts, and faults, all of which are configured from the JMX or SNMP management console. A sample MIB and MIB browser is included by default. - Cape Clear Orchestration Manager provides management and monitoring functionality for the Cape Clear Orchestration Engine. Using Orchestration Manager, an administrator can track the progress of business processes, view a history of business process activities, drill down to see message exchanges, and configure specific variables to be exposed within the console. - Any Cape Clear server, and any services deployed therein, can be managed via any JMX- or SNMP-aware management platform (for example, HP OpenView, CA Unicenter, or IBM Tivoli)

Category of the software product

Cape Clear is a “all in one” solution: it includes

- a process modeler (support of BPEL 1.1)

- a workflow engine
- a manager / monitor

Supported interfaces

Interface 1: Cape Clear 6.5 provides a graphical process designer integrated to Eclipse platform. It is possible to design business process, but no to import them.

Interface 2: Cape Clear 6.5 runs the BEPL process as a Web service. It is then possible to invoke the process from any client supporting web services (web clients, java stand-alone clients). Note that Cape Clear 6.5 provides a very useful tool inside the Cape Clear Studio: it can dynamically generate SOAP requests to test deployed business process. The screen is split in half: the first half contains the request that the developer can update with its own data (random data is generated by default). The second half is the SOAP answer from the service.

Interface 3: Invoked applications can be of any type as long as they provide a web service interface (i.e. a wsdl definition file). But not all platforms are service/WSDL-enabled, and not all data is XML Schema-aware or accessible. Cape Clear Studio offers a range of capabilities for creating interoperable and reusable endpoints from existing assets. This can be achieved by auto-generating WSDL interfaces from a variety of legacy systems. These WSDL-enabled entities are then accessible via transformation and mapping in the mediation layer.

Interface 4: Cape Clear 6.5 supports workflow collaboration: as the process is seen as a web service, it can be re used by any of other workflow systems supporting web services. Moreover, it's very easy to design an orchestration with the graphical process designers starting from the WSDL of collaborated workflows. Each partner can be identified by a given color in the diagram, which helps a lot for readability.

Interface 5: Monitoring business process can be done by the web application provided by Cape Clear: Orchestration manager. The access to the application is secured by a login/password. It gives a lot of information about processes, running processes instances. The server itself provides support for JMX and SNMP. It also provides a web-based application for the server administration (Cape Clear Manager) for stats, platform information, web services administration, etc.

Additional interfaces*:* Cape Clear provides support for long-running transactions by de (re)-hydrating the process instances. The status of running workflows can be stored in database to save memory and then brought back in the engine for execution. Cape Clear only supports Oracle, SQL server or any "JDBC aware" databases.

Cape Clear 6.5 also provides a support of UDDI. It is possible to retrieve a WSDL file by requesting a UDDI directory using a contextual tool (i.e. a search engine). The following diagram, excepted from a Cape Clear webminar, illustrates the possibilities in terms of interfaces, Figure 4.8.

Supported standards

Conformity to standards and exchange formats

a. Cape Clear 6.5 provides a full support to BPEL 1.1 specification. Activities such as Invoking, receiving, compensation, event handling, flow controls are

available in the process designer tool. It supports also WS-ReliableMessaging for transport reliability, and recovery.

b. Workflow execution tools: support and deviation from workflow standards; Technical environment is needed; Required software product (application servers, supported communication protocols, transaction mechanism bindings).

Applications can be deployed on jBOSS (recommended by the vendor), Tomcat, Weblogic, Websphere, Jonas.

The communication protocols supported are summarized in Figure 4.8: SOAP and non-SOAP messaging over HTTP(S), (S)FTP, SMTP (e-mail), WS-ReliableMessaging (WS-RM), File drop, JMS (bundled JMS server and WebSphere MQ).

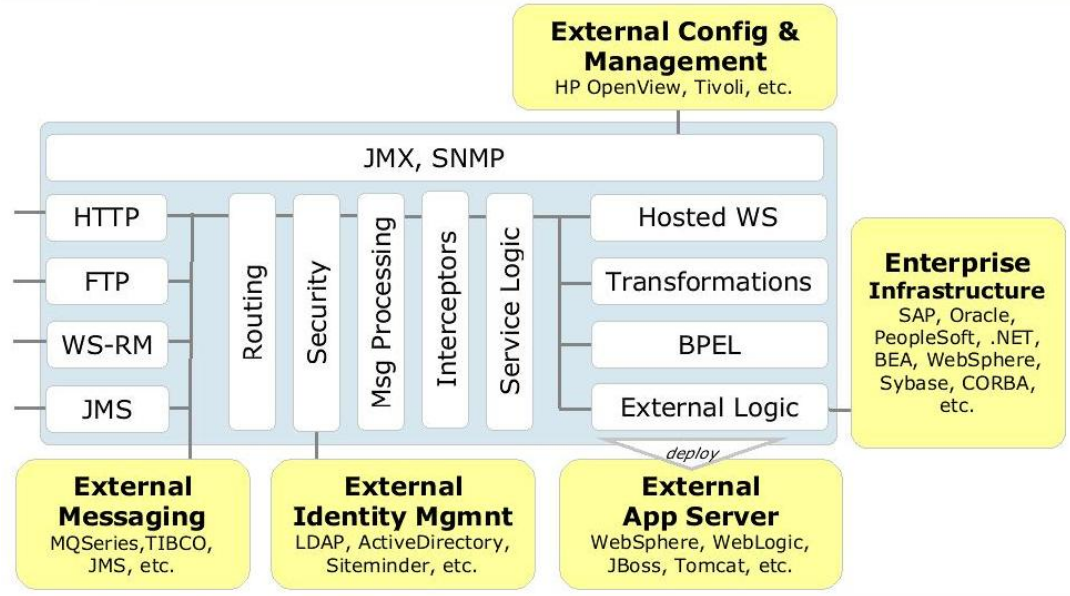


Figure 4.8 - Cape Clear 6.5' Interfaces

4.7. Con:Cern

Workflow software overview

Con: cern is a workflow engine based on an extended case handling approach [4.6].

Workflow software description

There are three tools included in con: cern:

1. a graphical modelling tool
2. a generator for process description, Java templates, unit tests and HTML documentation

Con: cern is a tool for the administration of processes and diagnostics.

Category of the software product

A process is described as a set of activities with pre- and postconditions. An activity is executed when its preconditions are met. It manipulates the process item, thereby creating postconditions. The process flow is determined at run-time.

The process is described via the modeler and then a Java code is generated. After the deployment of the process to an application server, a client can be written using the Java templates to connect to it.

Supported interfaces

Interface 1: There are no facilities to exchange of workflow descriptions (models, choreographies, orchestration) between the tool and other modelling tools, repositories, workflow execution tools.

Interface 2: Supports the invocation of workflow engines by workflow clients.

Interface 3: Implementation of the interface is done by using the standards used for deployment to a application server (JBoss-AS or Tomcat) and stubs are generated using the con: cern APIs. Notifications are not implemented.

Interface 4: Controllers can be deployed as MBeans (JMX Integration) or using a JTA transaction manager (Spring Integration).

Interface 5: Monitoring of the workflows activities is done via log files (in table form). There is no life cycle management of distributed workflows. A particularity is the fact that the graphical modelling tool can produce a timing Gantt chart if there are no cycles in the process.

Additional interfaces: There are no import/export facilities of data towards other products like databases, file systems, repositories (UDDI, LDAP); monitoring of transactions.

Supported standards

Conformity to standards and exchange formats

a. **Process definition tools**:

The process is defined using the graphical modelling tool allowing describing: activities, conditions, events, and a direct graph.
Complex workflows can be easily modeled.
Supported user interfaces: in order to describe the processes, the user can launch the local graphical modelling tool or to use its preferred browser to access the con: cern web site and activate the Modeler.jnlp.

b. **Workflow execution tools:**

Run on any J2EE-compliant application server. It is deployable in open source-only environment. Linux + SAP DB / Postgres + JBoss is recommended and tested. Since con:cern runs on a J2EE Server, it is possible to communicate with other software components, for example SOAP, JMS, JNDI, SMTP, CORBA, JDBC, J2EEE Connector Architecture (e.g. SAP).

4.8. Enhydra JaWE

Introduction

Enhydra JaWE (Java Workflow Editor) is an open source graphical Java workflow system that allows the user to create, manage and review process definitions (workflows) using a visual tool. It lets users create workflow process definitions, and store and reuse them later. JaWE has been developed fully according to WfMC specifications supporting **XPDL** (XML Process Definition Language) as its native file format and LDAP connections. Thus, it can be used to edit/view any XPDL file that conforms to WfMC specifications. The final XPDL file produced from using this tool can be interpreted at runtime by a workflow engine (Enhydra Shark). Because of involving the WfMC proposed package concept, JaWE divides itself into two logical parts: **Package Level** and **Process Level**. As will be explained in detail below, the Package level manages entities and attributes within the Package, while Process level manages entities and attributes within Workflow Process Definition.

JaWE is a tool for ***Process Definition*** modelling. The final output of this process modelling is a XPDL output file, which can be interpreted at runtime by the workflow engine(s). JaWE accomplished three main goals:

- Graphical representation of process definition
- Export of process definitions to XPDL
- Import of any valid XPDL and its graphical representation

A workflow process definition, generated by JaWE, is capable of interpretation in different workflow run-time products. The principles of ***Process Definition Interchange*** are based on Meta-Model framework, which identifies commonly used entities within a process definition, their relationships and attributes. A variety of attributes describes the characteristics of this limited set of entities. Using this Meta-Model, JaWE can transfer models using a XPDL as a common exchange format. Beside this interchange, JaWE is also used for internal representation of process definitions. The whole concept is shown in Figure 4.9.

There is a mandatory minimum set of objects, which must be supported within XPDL. This "minimum meta-data model" identifies those commonly used entities within a process definition and describes their usage semantics. Extensibility is provided by the facility to encompass additional object attributes ("extended attributes") which can be included as extensions to the basic Meta-Model to meet the specific needs of an individual product or workflow system.

Meta-Model

The Meta-Model (Process Meta-Model) describes the top-level entities contained within a Process Definition, their relationships and attributes. It also defines various conventions for grouping process definitions into related process models and the use of common definition data across a number of different process definitions or models.

Here is a short list of the entities:

- *Workflow Process Definition*
- *Workflow Process Activity*

- *Transition Information*
- *Workflow Participant Declaration*
- *Workflow Application Declaration*
- *Workflow Relevant Data*
- *System&Environmental Data*
- *Resource Repository or Organizational Model*
- *Data Types and Expressions*

All of these entities are maintained by JaWE, except "System & Environmental Data".

JaWE gives two logical parts of application as said in the previous text. There are "Package level" and "Process level". Package level manages entities and attributes within Package while Process level manages entities and attributes within Workflow Process Definition. According to this separation, Tutorial is organized in the way to explain these two parts of JaWE according to WfMC.

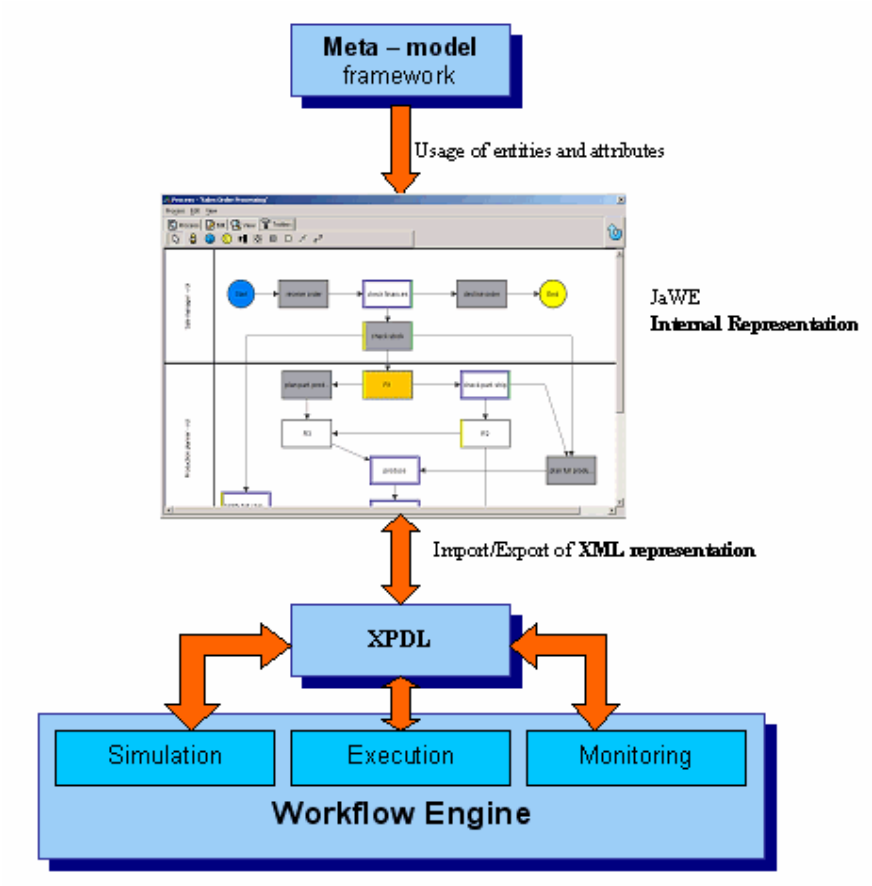


Figure 4.9 - Concept of the Process Definition Interchange

Package level

A view of Package level is shown in Figure 4.10.

At the package level, the user is allowed to choose among several XPDL views such as graphical, text, and xpdl view, which represent the XPDL, as it will be saved into file. Graphical view of package level is divided into some parts. Left side of the main window shows Package Hierarchy Tree. Root of the tree is main package and branches are imported external packages. If these external packages have their own external packages, they are shown in the tree too. Since all packages have unique IDs, circular references are allowed, though tree expansion will stop on package, which is displayed before. Right side of the main window displays workflow processes defined in package, which is selected on the hierarchy tree.

Process level

The second part of JaWE is Process level, which is some kind of editing window. This part of JaWE is used for graphical representation of process definition and for defining attributes of entities on that level. Figure 4.11 represents this level:

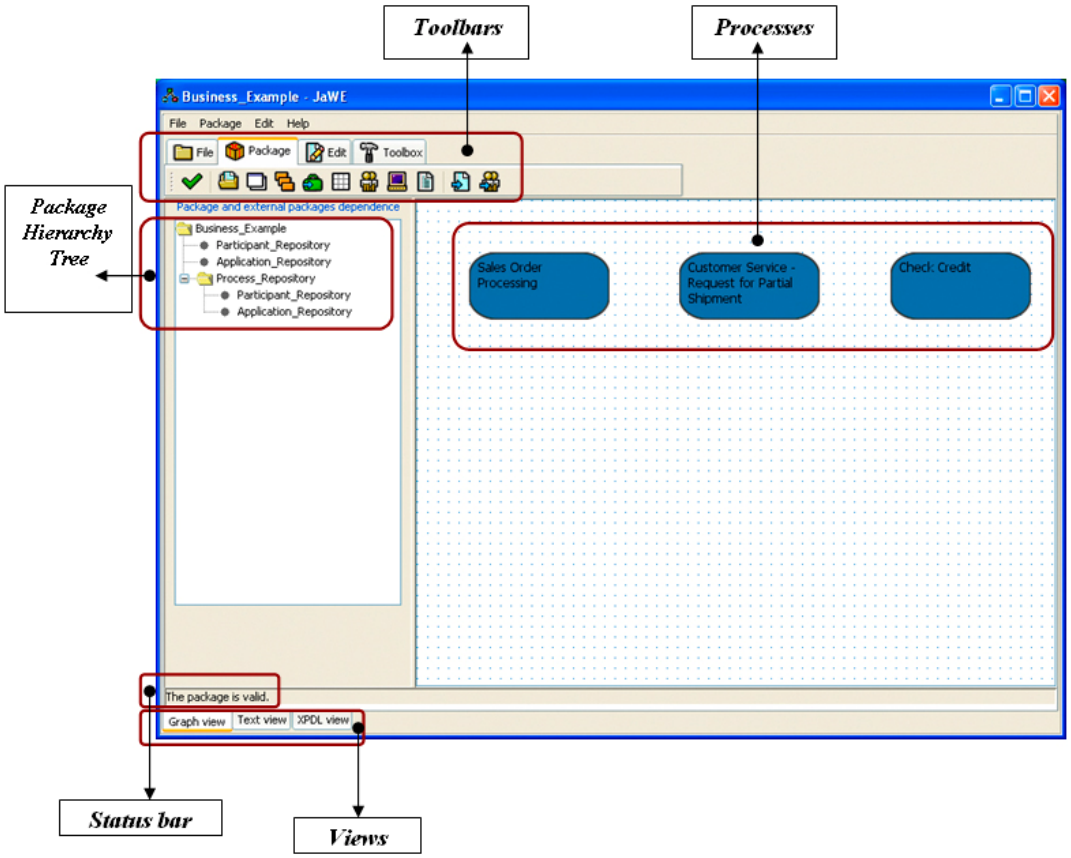


Figure 4.10 - Package level

The Process level view is essentially an editing window that can be used for graphical representation of process definition and for defining attributes of entities on that level. In the visible working area, the user inserts visible objects and adjusts them. The first thing drawn must be a participant, after which the user may insert other elements such as Activities and Transitions.

There are four kinds of Activities defined in JaWE:

- Generic Activity
- Sub-Flow Activity
- Block Activity
- Route Activity

The link between two activities is established by transitions. Transitions are more than just link between activities. They describe possible transitions between activities and the conditions that enable or disable them during workflow execution. JaWE has three types of transition - simple, self-routed and circular. Simple transition is link between two activities, represented graphically with one straight line. Self-routed transition is link between two activities which is graphically 'broken' in three parts (but essentially, they do not differ in the XPDL logic they are representing), and circular transition is transition from activity to itself, and is graphically represented as a circle with an arrow.

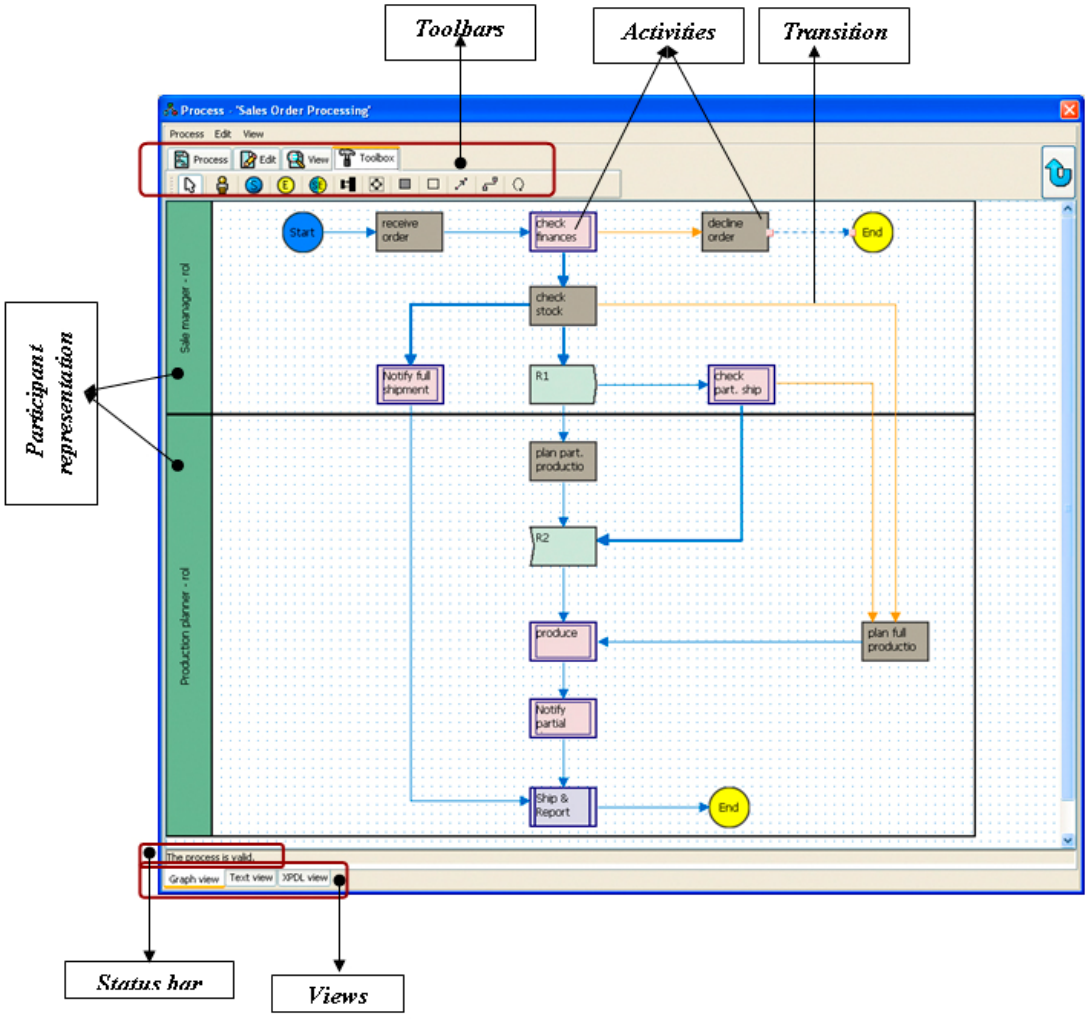


Figure 4.11 - Process level

Text view

Text view is used for showing a tree-like structure of designed XPDL. In some cases, it can make the editing of XPDL quicker, because when the user right-click appropriate element in tree-structure, he get the dialog for editing its properties. It is also used to display XPDL in more user-friendly way, and it provides action to save this user-friendly description of XPDL. In addition, there is a Print button in the menu, but printing is not implemented yet.

XPDL view

XPDL view is used for showing a structure of designed process definition in form of XML (or, more precisely, in XPDL). Every entity or attribute, which is graphically added or changed in JaWE, is automatically shown in this panel - XPDL view.

4.9. Enhydra Shark - Java Open Source XPDL Workflow

It is a workflow engine completely based on WfMC and OMG specifications [4.7].

Enhydra Shark architecture
Based on the agreed components based approach the current architectural goal of Enhydra Shark looks like the following, Figure 4.12 [4.8].

Shark Admin application is Java swing application meant to be used by administrator to manage Shark engine. There are two kind of admin application, the first one is using shark directly as a library, and the other one communicates with shark deployed as a CORBA service using shark's CORBA wrapper interface. It can be used to handle shark's external repository containing XPDLs (to upload new XPDL files or delete existing ones), to load some XPDL file into shark, unload it, update it, to instantiate and monitor shark's processes, to perform mappings among participant definitions and real users, and among application definitions and Tool agents. It also contains a built-in worklist handler application that can be used for performing workitems, or for reassigning workitems from one user to another.

The process monitor is divided into four major parts, Figure 4.13. The package-process definition-process instances tree enables to select a running instance of a package's process definition. When the user selects the process instance, other parts graphical data correspond to this process instance. He can see the main properties of the instance (the name, and the current state), the graphical diagram of the process instance with activities that are currently running being marked, to perform different operations on that process instance using the buttons at the bottom.

The operations, which can be performed, are:

- start the process;
- suspend the process;
- resume the process;
- terminate the process;
- abort the process;
- view the process history;
- see the description of the process;
- see and edit the process variables, and that way the user can manage the process flow if needed;

- Enter activity management dialog. The dialog displays the list of process activities, and when the user selects one of them, its current state is displayed in the text box.

From this dialog, the user can perform additional operations on single activities:

  - suspend activity
  - resume activity
  - terminate activity abort activity - process becomes 'stucked'
  - manually start an activity
- delete all finished processes
- delete selected process
- perform a check for activity deadlines for all processes
- perform a check for limits of all processes and activities

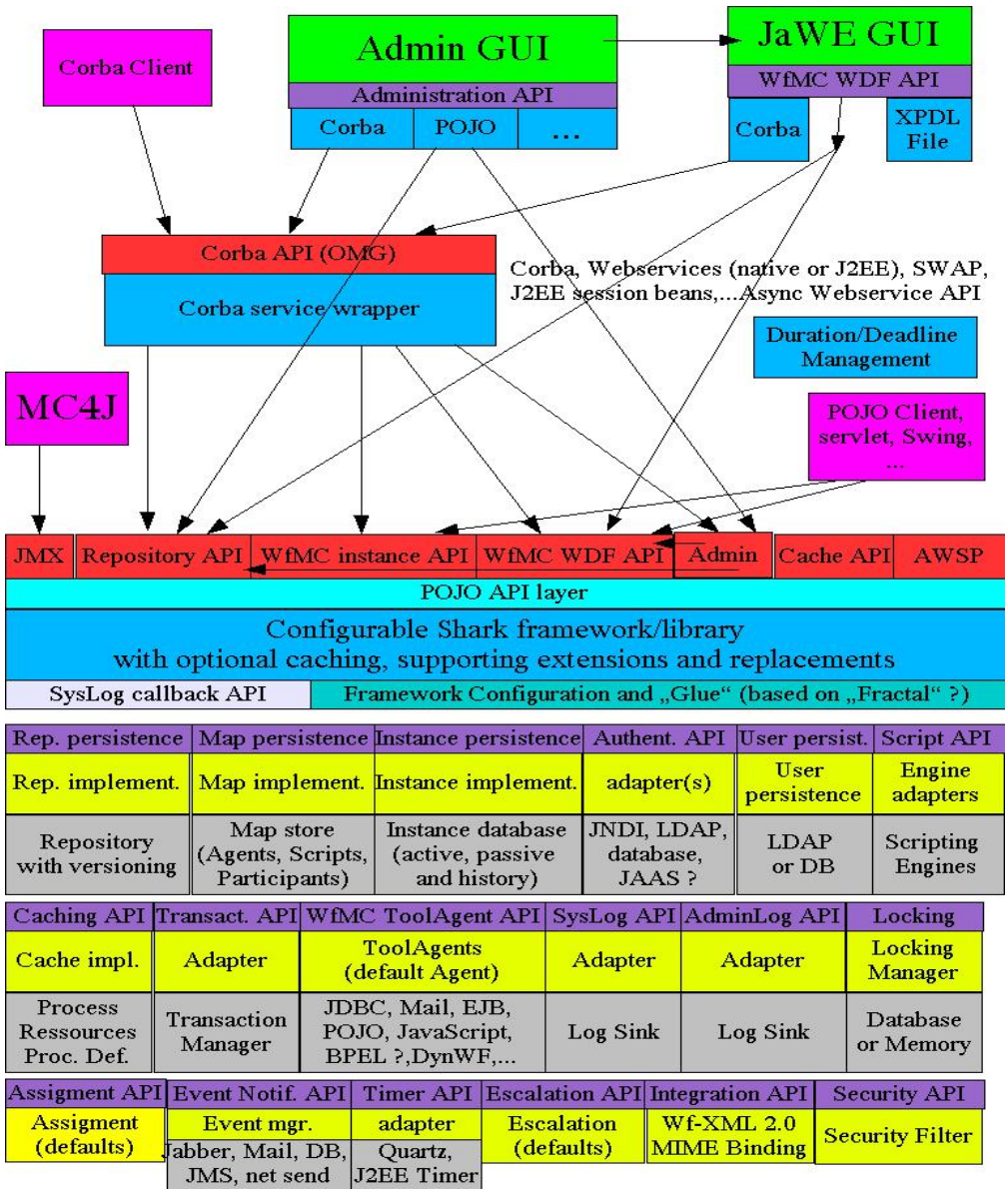


Figure 4.12 - Architecture of Enhydra Shark

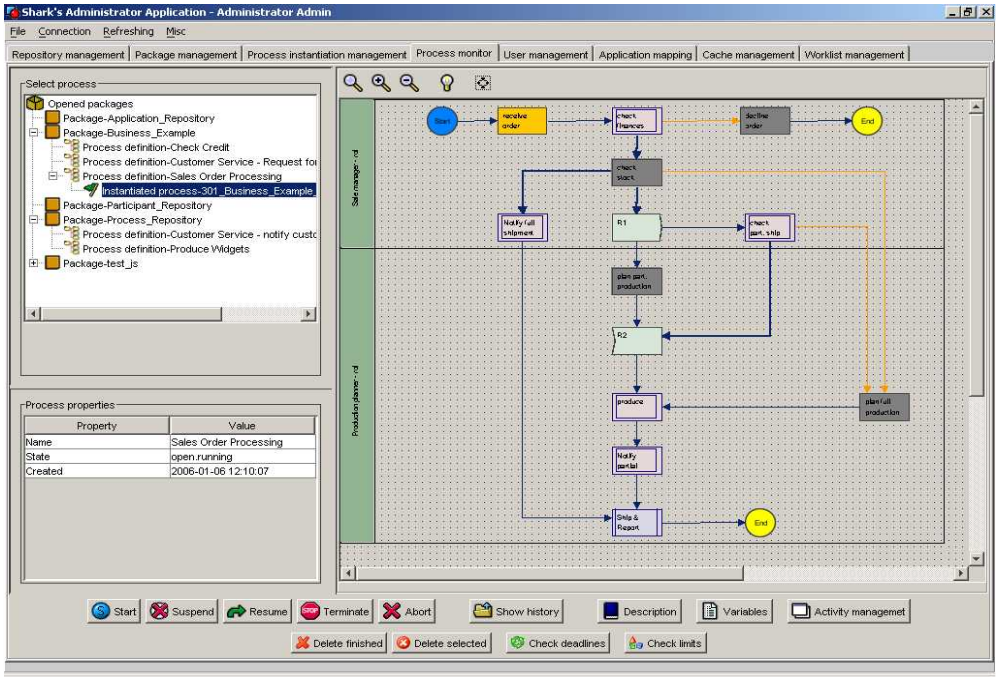


Figure 4.13 - Enhydra Shark process monitor

The user management console's page is divided into three parts. Functionalities:

- Accounts – the administrator can manage the users of the shark server by defining the new ones, deleting the existing ones or changing their properties.
- Logged - displays the list of currently logged users
- Mapping - enables the administrator to map the package and package's processes participants to the real shark users.

Enhydra Shark allows mapping a package and package's processes applications to the real applications handled by a tool agent. Currently, six tool agents come with the Shark distribution. To map application definition to tool agent application, the user has to go to the application mapping section of admin application, and press the "add" button. The dialog will arise, and one has to select the application definition at the left side of dialog, and the tool agent on the right side of the dialog. Then the user should enter some mapping parameters for tool agent. When a mapping of the application definition to the tool agent is done, shark will try to connect to the proper tool agent and ask him to execute its application, and will retrieve the results of execution. Here is the brief description of parameters that the user can enter when mapping of the application is performed:

- User name and password - not required for tool agents distributed with Shark. Some other tool agents can use it when calling applications that require login procedure
- Application name - the name of application that should be started by tool agent (i.e. for JavaClassToolAgent that would be the full name of the class, for RuntimeApplicationToolAgent it would be the name of executable file that should be in the path of the machine where tool agent resides. For JavaScriptToolAgent this can be either the name of the java script file, or the java script itself, depending on Application mode attribute.), for SOAPToolAgent it is the location of WEB service

and for MailToolAgent it is a class of MailMessageHandler called to actually send/receive mails.

- Application mode - various tool agents use this attribute for different purposes. The RuntimeApplicationToolAgent uses mode 0 to indicate that it should not finish execution until the system application is finished (otherwise it will start system application and return finished status -> activity does not wait for system application to finish, but process proceeds to the next activity), and JavaScriptToolAgent uses mode 0 to indicate that it should search for java script file .

Enhydra Shark project delivers a workflow server with a difference. It is an extendible workflow engine framework including a standard implementation completely based on WfMC specifications using XPDL (without any proprietary extensions) as its native workflow process definition format and the WfMC "ToolAgents" API for serverside execution of system activities. Every single component (persistence layer, transaction manager, scripting engines, process repository,...) can be used with its standard implementation or extended/replaced by project specific modules. This way Enhydra Shark can be used as a simple "Java library" in servlet or swing applications or running in a J2EE container supporting a session beans API, Corba ORB or accessed as a web service.

4.10. EnterpriseXtention

Workflow software overview

Workflow software presentation

EnterpriseXtention is a comprehensive Document & Workflow management system consisting of two main elements – Document Repository **x.File** and Workflow Engine x.Flow, and a number of „surrounding" applications (like Corporate Mail Room x.MailRoom or Project Manager x.Project).

X.Flow is a Production Type Workflow Engine based on XPDL specification (using prioprietary extensions).

X.Flow engine analyses each Business Process instance, interprets its model (definition), stored in the process model database and resolves activity (workitem) transition and participants, based on organization model stored in the organization model database.

Current version of enterpriseXtention (2.52, released in October 2005) has its only implementation on IBM Lotus Domino / Notes platform.

Forthcoming version of X.Flow (3.0, release planned for June 2006) will be platform independent J2EE product based on SOA. The evaluation will concern the 2.52 version. The 3.0 version's planned features will be described in the comments

Workflow software description

Workflow software consists of workflow engine, processes and activities storage, activities user interface (worklist), process models storage and additional elements such as organization model module.

Process model is stored as a set of Notes database documents for processes, activities and transitions. Database contains also a set of user tools that can be included into activities to achieve needed functionality. Process model includes functionality defined in WFMC standards and can be translated into XPDL (XML) files.

Workflow engine due to process model starts process instances, creates activities and changes statuses for activities marked as done by users.

The engine is implemented as an agent in Lotus Notes database. It runs every "n" minute (the default setting is 5 minutes interval) and searches for activities that are ready to process by the engine. A separate element of the engine is Access Flow module (to update access control list of the documents involved in the processes).

User Interface for accessing activities is implemented for Lotus Notes Clients and contains a user activities' (worklist) view, which shows activities to be performed by the user. Every activity is displayed with tools defined in model.
Authorized users with higher access can inspect flow of processes that they had taken in part.

User interface is integrated with document repository to display documents associated with current activity. Documents can be organized into projects and folders in another module (X.File).

Category of the software product

The main functionalities of X.Flow include:

- ✓ Modelling, validating and simulating processes
- ✓ Modelling of the organization structure (chart) to resolve workflow participants according to their functional or hierarchical roles or organization units, or to resolve possible proxies for main participants
- ✓ Processing and monitoring Business process instances
- ✓ Delivering activities (workitems) to the workflow participants via worklist or directly to participant's mailboxes
- ✓ Providing automatic modification of access control list of the related documents
- ✓ Providing graphic tools for modelling: Version 2.52 uses Java Organization Editor – TRIDENT Software Group product – for Organization model; in the case of the process model, Open Source JaWE can be used. Version 3.0 will use the complete GraphicToXML tool TRIPLAN v. 2.0 by TRIDENT Software Group.
- ✓ Providing graphic viewer of the process status (Gantt chart style) by means of TRIPLAN v. 1.2
- ✓ Providing user graphic interface for workflow participants, integrated with the other enterpriseXtention modules

Providing WEB access (MSIE) to the worklist

Supported interfaces

Interface 1: Exchange of workflow descriptions (models, choreographies, orchestration) between the tool and other modelling tools, repositories, workflow execution tools.

Process models can be exported as xpdl (xml) documents. Some activity procesing tools assignments use extensions to xpdl specification (extended attributes).

Interface 2: Supports the invocation of workflow engines by workflow clients; Implementation of the interface (by WSDL interface or a programming language interface).

Invocation of workflow processes is possible via Lotus Notes client (full functionality) or WEB client (MSIE) with the above limitations

Interface 3: Application invocation, which perform additional workflow activities; Implementation of the interface (by WSDL interface, programming language interface; notification mechanism for human actions).
For additional workflow applications and complex activities, invocation X.Flow uses proprietary "activity beans" defined in the process definition (model).
WSDL interface available for additional workflow applications invocation in v. 3.0.

Interface 4: Support of workflow integration and collaboration (between different domains and/or distributed workflow execution platforms.
Organization model is designed to support B2B workflow applications between different domains.
Distributed workflows will be available via WSDL interface according to v. 3.0.

Interface 5: Monitoring and administration of the workflows activities; Life cycle management of distributed workflows.

Processes can be modified in such a way that all started processes run according to modified model or can run according to completely new model.
Models can be marked as archived.

Additional interfaces: import/export of data towards other products like databases, file systems, repositories (UDDI, LDAP); monitoring of transactions.

X.Flow stores all process data in the Lotus Notes database. All these data can be exported in XML format.

Organization Editor JOE can use LDAP service to will units and roles with real people data.

Supported standards

X.Flow uses XPDL as process definition language with some proprietary extensions (extended attributes are used). Processes can be modeled manually in the text or import mode via heavy user /v. 2.52/ or graphically, as described in the section G 1.3. /v. 3.0/ lightweight browser will be used as well.

IBM Lotus Domino server is the only platform for workflow execution /v. 2.52/. Every Web Application Server, supporting J2EE /v. 3.0/.

X.Flow can execute every process, defined according to XPDL specification.

4.11. Freefluo

Workflow software overview

Workflow software presentation
Freefluo is a workflow orchestration tool for web services initially developed by IT Innovation but now available to all from the Freefluo Sourceforge Site. It can handle WSDL based web service invocation. It supports two XML workflow languages, one based on IBM's WSFL and another named XScufl that is under development as part of the Taverna

Sourceforge project [4.9]. Freefluo is very flexible, at its core is a reusable orchestration framework that is not tied to any workflow language or execution architecture. Freefluo includes extension libraries that enable execution of workflows written in a subset of WSFL. Support exists for discovery via standard UDDI and recording of provenance.

Workflow software description

The engine has a basic command-line client (primarily for testing) and is very easy to deploy as a web service to provide multi-user access to workflow execution services. It contains libraries that enable execution of workflows written in a subset of IBM WSFL and in Scufl (Simple Conceptual Unified Flow Language). Scufl (and XScufl, its XML representation) was developed by It Innovations together with some member of myGrid consortium [4.10].

The graphical environment for building workflows is not included with Freefluo, while it is provided from the Taverna project. Freefluo is bundled as an embedded workflow engine with Taverna.

Main functionalities

The Freefluo workflow enactment engine is a task orchestration tool and interpreter for executing workflow scripts. Using Freefluo, it's possible to execute workflows to coordinate web services, local applications and your own custom processing steps.
It supports Scufl workflow specifications and a subset of IBM's WSFL (Web Services Flow Language).

Supported interfaces

Interface 1: Exchange of workflow descriptions (models, choreographies, orchestration) between the tool and other modelling tools, repositories, workflow execution tools.
The workflow engine supports execution of workflow scripts written in the Scufl workflow language. Scufl is being developed as part of the Taverna project. The tool can also execute a subset of WSFL.

Interface 2: Supports the invocation of workflow engines by workflow clients. Implementation of the interface is by WSDL interface or a programming language interface.
The engine can be called by web service (SOAP). In fact, the engine exposes some methods described by a WSDL file.

Interface 3: Application invocation, which performs additional workflow activities. Implementation of the interface is by WSDL interface, programming language interface; notification mechanism for human actions.

WSFL is an XML language for the description of Web Services compositions, so external applications can be invoked by Web Services interfaces.

To define the public interface of the composition, the <flowModel> WSFL element includes a declaration of the supported service provider type as an attribute of the flow model. This mapping is specified by an <export> element, which relates an activity of the flow model and an operation of its public interface. The mapping defines the effect of each operation by relating it to the execution of the internal composition.

As indicated in the figure, the relation between an operation on the public interface of a flow model and an operation of a service provider is established through a <plugLink> element. Thus, a plug link represents the inherent client/server structure of a Web Service.

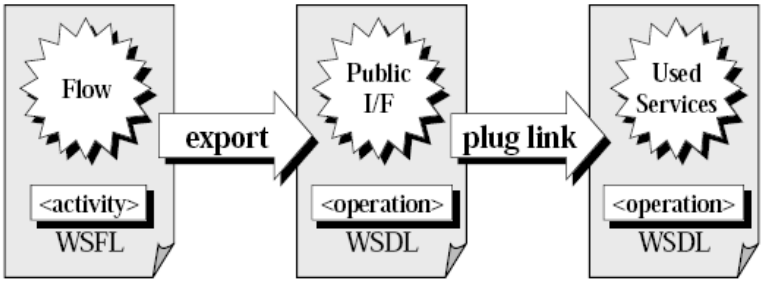


Figure 4.14 - Supported interfaces

Interface 4: Support of workflow integration and collaboration (between different domains and/or distributed workflow execution platforms).

Interface 5: Monitoring and administration of the workflows activities; Life cycle management of distributed workflows.
No tool is provided with engine distribution. In fact, the only way to have monitoring and administration tools is using the Taverna workbench, Figure 4.15.

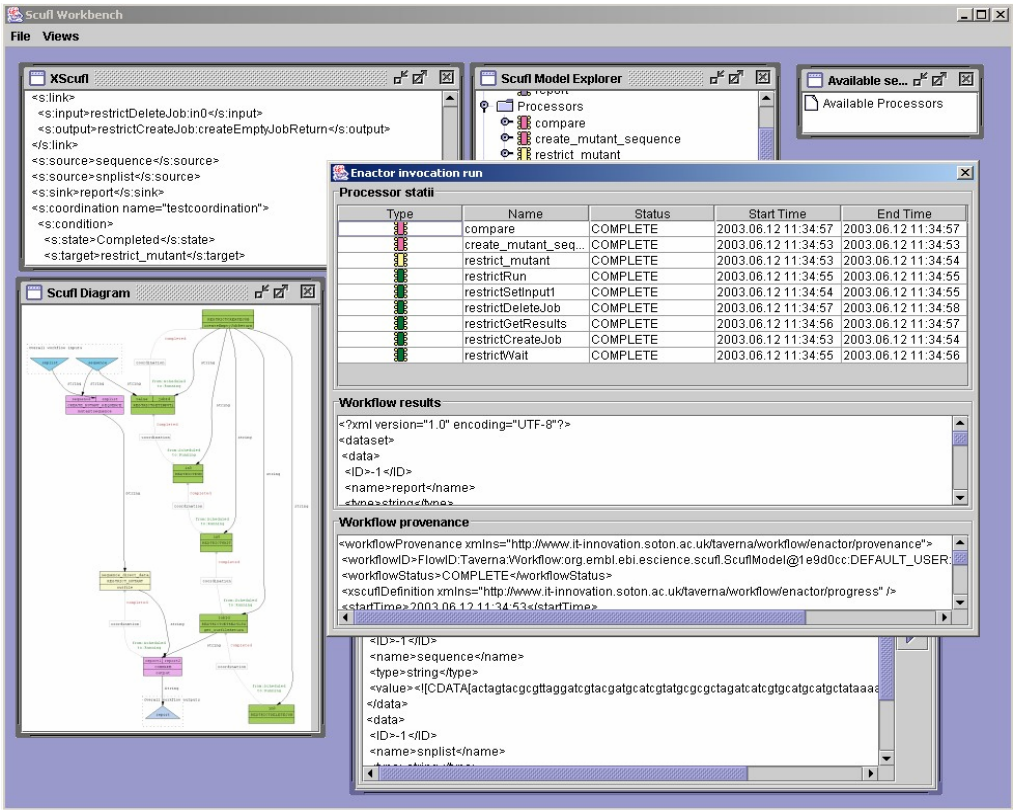


Figure 4.15 - Taverna workbench

Additional interfaces: import/export of data towards other products like databases, file systems, repositories (UDDI, LDAP); monitoring of transactions.
The engine supports discovery via standard UDDI.

Supported standards
The engine supports SCUFL and WSDL. Security mechanisms between clients and FreeFluo web system are provided using HTTPS protocol.

4.12. JBoss jBPM

Workflow software overview

Workflow software presentation

jBossjBPM is java based BPM Open Source solution. It can run on any J2SE compliant architecture. Latest version available is 3.0.2 under a LGPL licence.

Workflow software description

jBpm mainly consists of two parts :
- a XML file defining the workflow chart (jPDL)
- a JAVA API to implement nodes and transitions

jPDL description. Consider your application as two parts : 1) a graph of a state machine which represents the business model, business interactions, states, etc. and 2) the code (JAVA code) to implement a transition to a state, or implementing task.

The graph: The graph is just a graphical representation of an XML file (jPDL) which is convenient for business analysists. A graphical process designer can be downloaded from jBoss (under LGPL licence). It is very easy for "non coding people" to use it and design process models. Every graph must start with a "start state" and end with an "end state". The jPDL XML file is generated, as well as a picture file of the diagram, which is convenient to import in document or just check. Another xml file is generated (gpd.xml), which is the textual representation of the graph, giving coordinates of nodes, transitions etc. An example of a graph is given in Figure 4.16.

All these three files are enclosed in a "Process Archive" (.par file).

The code: It's just all the Java code that will implement your tasks defined in the jPDL file and make you go forward on your graph. The graph editor enables you to choose which class is going to implement a Task or a transition just by right-clicking. This is possible due to interfaces defined in jBPM that classes must implement (for instance ActionHandler).
Building application on top of Java systems makes it very easy to integrate and deploying over J2EE gives a lot of possibilities in terms of interactivity with the "outside world" (web services for instance).

As some transactions can take a lot of time (for example waiting for a client interaction, or shipping task), jBPM stores the definition, the execution (instance of a definition), and the logging in a database. jBPM supports the main database systems such as MsSQL, MySql, Oracle. Default database used is hqsldb (Hypersonic), as it can run in an "in memory" mode

that does not write anything on disk and whose schema is always fresh as it's rebuilded every time it's deployed.

One more interesting thing is the integration of Hibernate 3.0. This means that hibernate is responsible for the conversions between the jBPM java objects and the persistent representation of these objects in a relational database. Though it doesn't mean to have any knowledge in Hibernate to build a jBPM based application as it's only used internally by jBPM engine. The database schema is provided in jBPM package, and the only configuration to be done is to give the database location information as well as credentials and database type, Figure 4.16.

Category of the software product

jBpm is both a modeler and not really an engine but a library that can be run in a JAVA engine (aka JVM). The modeler part is provided by an Eclipse graphical plugin. You can use any JAVA editor to implement your workflow.

Supported interfaces

Interface 1: The process definition is stored in a jBPM database (Oracle, MySQL on which a jBPM schema is created).

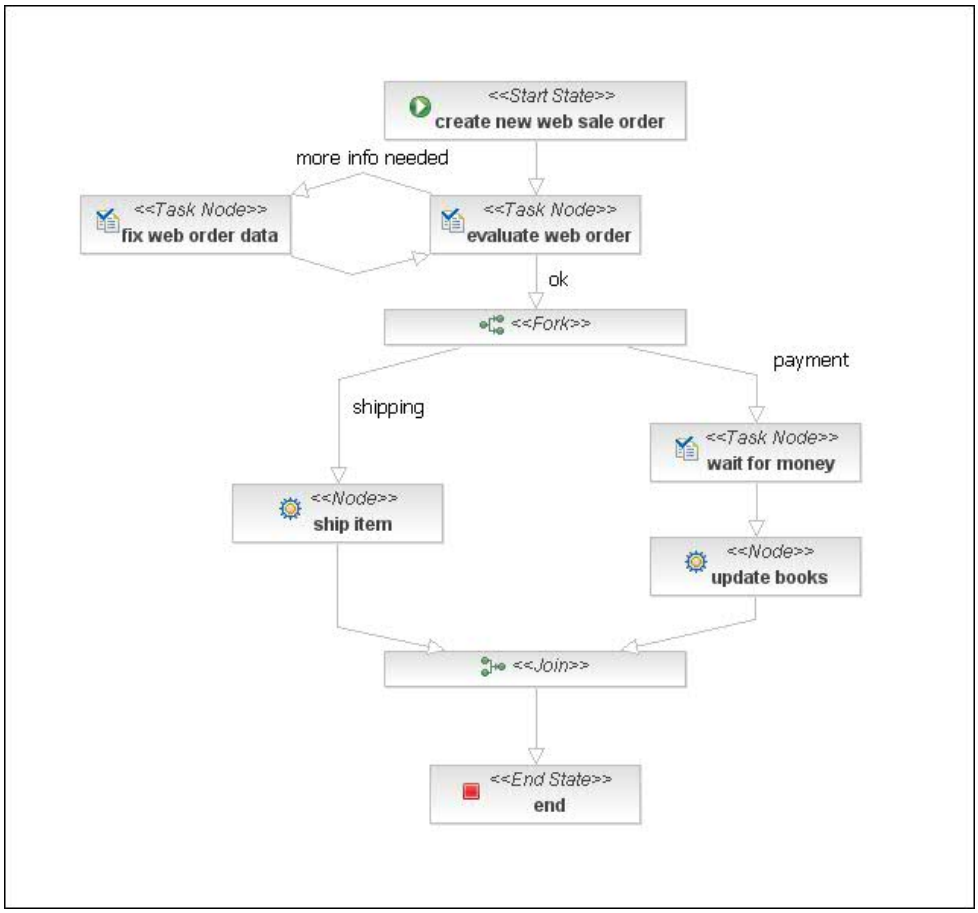


Figure 4.16 - Example of graph from the editor

Interface 2: Any kind of clients as long as the application has set its hooks. For example, the application can be seen as a web service, but a mail client or web client can also interact with it.

Interface 3: can send mail, call a web service (and thus any kind of application)

Interface 4: Can be implemented with web services.

Interface 5: Workflow activity is persisted in database, so any database client can retrieve or set data. A web application is also provided.

Additional interfaces: Support of any kind of relational database (Oracle, MySQL) as well as repositories.

Supported standards

Conformity to standards and exchange formats

a)Process definition tools: the plugin tool supports jPDL only. It is not designed to model complex workflows. This plugin is dedicated to Eclipse platform.

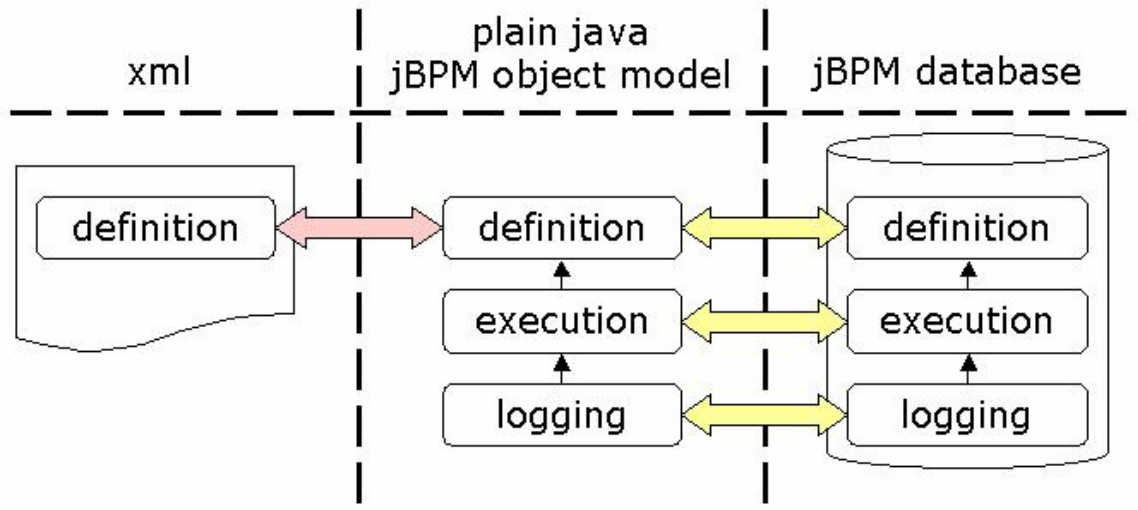


Figure 4.17 - jBPM database overview

b) Workflow execution tools: The workflow can be run in any J2SE context: as a JAVA standalone application, web application, SWING, EJB. The plug-in requires Eclipse 3.0+ and J2SE 1.4. JBPM is dependent from other libraries such as Hibernate (for O/R mapping, de-hydrating), dom4J (xml parsing), CommonsLoging and Beanshell (light JAVA interpreter). As it can run on any J2SE environment, you can deploy the application on a web application server (jBOSS, Tomcat). The supported protocols are all the ones that can be used in a Java environment (like SOAP/HTTP). Transactions are supported in the jBPM API (begin Transaction (), commitTransaction () and rollbackTransaction () called on jBpmSession object.

4.13. ObjectWeb BONITA

BONITA (based on WfMC XPDL standard) is a workflow system featuring a lot of innovative features like activities that can start in anticipation, awareness infrastructure allowing users to be notified of any events occurring during the execution in a given process , or automatic activation of user's code according to a defined activity life cycle [4.12]. Traditional workflow features like dynamic user/roles resolution, activity performer and sequential execution are also included in Bonita to support both cooperative and administrative workflow processes.

BONITA is a fully conformant J2EE application, taking advantage of the power and robustness of the J2EE platform. The BONITA API is accessible either through EJB's. Processes are created using a graphical definition tool or by using the Project interface API. A process is defined as a set of activities and an associated execution model. The enactment engine takes care of scheduling the activities according to the defined execution model. The User API provides full control over the execution of the process, for example allowing starting or stopping an activity. BONITA supports also dynamic modification of an existing process that is the Project interface can be used against a running process.

The Bonita system is composed of two main components or sub-systems:
- The Bonita Modelling Component
- The Bonita Execution Component

Bonita Modelling Components

Terminology
- A **process** is a set of activities. In BONITA, the term project is also used.
- An **activity** is an atomic unit of work. In BONITA, activities are also termed **Nodes**.
-A **transition** is a dependency expressing an order constraint between two activities. In BONITA, transitions are also termed **Edges**.
- A **property** is a workflow unit of data, commonly known as workflow relevant data.
- A **hook** is a user-defined logic adding automatic behaviour to activities/nodes.
- A **mapper** is a unit of work allowing dynamically roles resolution at workflow instantiation time.
- A **performer assignment** is a unit of work adding additional activity assignment rules at run time.

The Workflow Modelling Component (GraphEditor) allows the user to represent and visualize workflow processes. This tool allows the user to create and edit workflow processes using the graphical user interface, Figure 4.18.

Workflow processes are constructed by chaining basic functional units, called **Activities**, which can be defined and edited by the GraphEditor tool. Attributes of these activities, such as activity type, activity description, activity deadline, activity role, etc. can be modified by the user. Activities such as these can be connected using **Edges**.

The Bonita modelling component allows the definition and visualization of the process by different users at run time, and it thus involves some coordination and groupware methods for better communication between users.

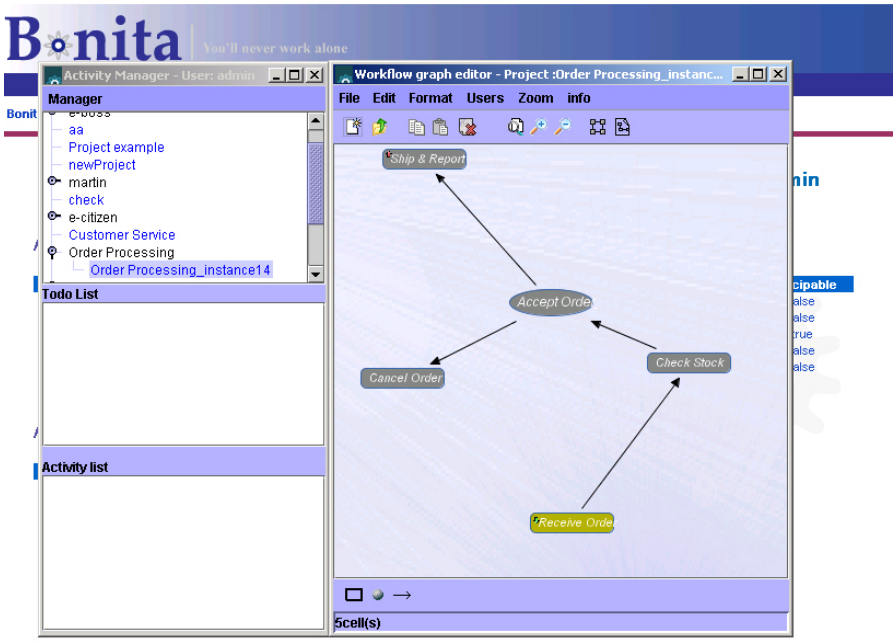


Figure 4.18 - Bonita GraphEditor tool

The Bonita Execution Component - is based on the new activity anticipation model proposed by the ECOO team [4.13].

The Bonita Execution environment is based on the principles of flexible data and workflow management, which allows activities to exchange data more efficiently. The Bonita system offers a WorkList application, which allows the user to control process execution. The user WorkList provides different information about the projects of every user. The WorkList is organized in three different lists: project list, todo list, and activity list.

The Bonita Execution Engine allows activities to share intermediate results when executing, thus executing in a non-isolated way. The workflow execution is based on the principle of anticipation, which allows an activity to escape to the traditional start-end synchronization model such that the user can cancel an execution activity or change the activity execution mode at runtime.

The Bonita Workflow also includes a browser-based environment with Web Services integration (using SOAP and XML) that allows users to define and control the workflow processes by means of a browser.

The basic process implementation in Bonita is the Project CMP Bean that represents the key component of the Bonita Cooperative Workflow Kernel. Every workflow process contains data information (process name, user creator, etc.) and associated data in terms of activities, connections between these activities, process data, process properties, and participants.

Figure 4.19 shows the basic Architecture of the Bonita System.

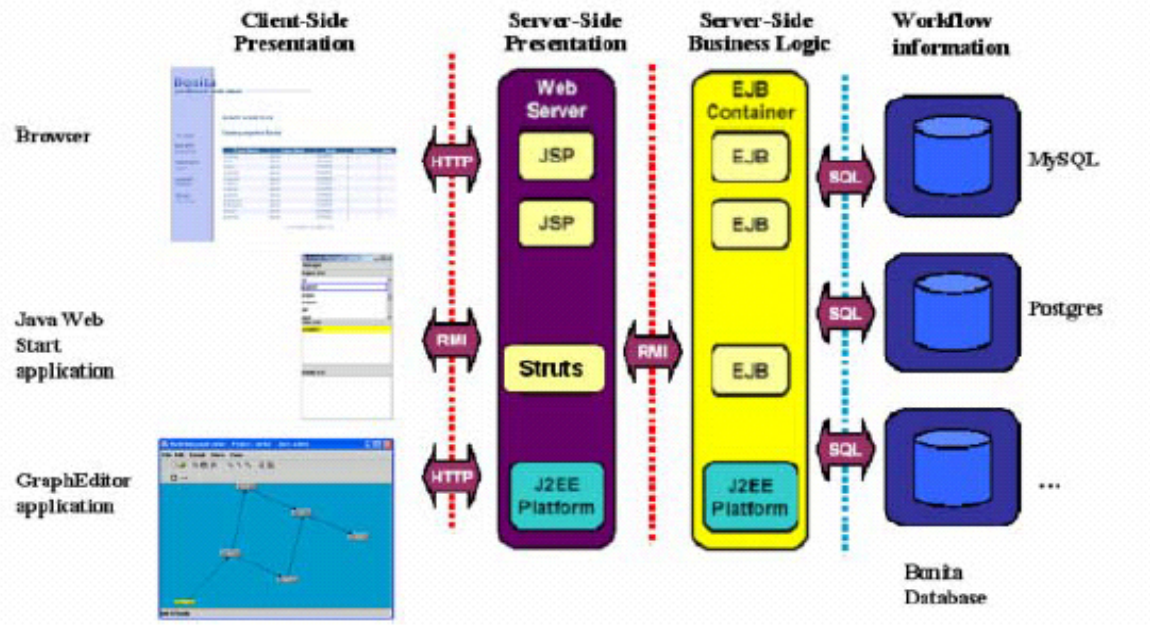


Figure 4.19 - The Bonita Architecture

Since the Bonita system is meant to be used in a cooperative environment, it includes some additional features such as the ones described below.

- **The Bonita JMS message service** implementation: notifies the definition and execution changes within a workflow process. Every user interaction is notified to Bonita Kernel and it is throwing a JMS event.
- **Bonita activity deadline service** - uses the Java Management Extensions (JMX) to advice the user if an activity does not end at the expected date.
- **Bonita Jabber** service notification that allows the users to receive notifications at real-time and exchange different kinds of messages.

Some Usability Issues

- The Bonita system does not include very sophisticated computational steering facilities, as a result of which, the user does not have complete control over the workflow as it is executing.
- The Bonita system does not include extensive data-mapping and data transformation features. The ones that exist thus far are very basic.
- Since the Bonita system is developed with Business Organizations in mind, it does not handle scientific data very accurately. It also has limited visualization features.
- The Bonita system includes very basic decision-supporting features, and the user has limited ability to define decisions and conditions on his own.
- The Bonita system does not provide extensive verification and validation features while constructing the workflows. If an exception is caught during execution, the system displays error messages on the screen that may not be completely understood by the user.
- The Bonita system does not allow easy interoperability with other systems.

- Workflow Administrators cannot view, modify and update certain project information at this point.
- The workflows described in the Bonita system do not conform to any particular standard, so the user is not able to edit the workflow for minor changes in text format.
- The current version is pre-configured to with the JonAS Application server, and to change it to work with an application server of the user's choice is not a trivial task.

4.14. OpenWFE

Workflow software overview

Workflow software presentation

OpenWFE is an Open source WorkFlow Environment written in java. It is a complete Business Process Management suite.

Workflow software description

OpenWFE features five components: a workflow **engine**, a **worklist** handler, an **apre** (Automatic Participant Runtime Environment), **webclient** application and web interface for modelling business processes (**Droflo**).

Engine interprets and runs process definitions. A Business Process Definition describes the transition of Workitems between participants. A Participant is usually mapped to a role in an organization. Engine can run many instances of each of those definitions.

The engine relies mainly on 2 services:
- The Participant Map contains information about how to dispatch workitems to the Participants,
- A Listener 'listens' for workitems coming back from participants.

A **Worklist** is a set of Workitem Stores.

There is usually one store per participant (per role). The role is 'performed' by a human (a user) or by an automated participant (**apre**). A Workitem, depending on the flow, could represent a task, a set of files, a request, etc.

The Worklist component exposes a Worksession interface that enables its clients to manipulate the workitems in its stores. This worksession also contains methods for launching processes (creating running workflow instances from workflow definitions).

Webclient gives the access to the worklist. OpenWFE can run completely standalone without any Application Server, requiring only a Java SDK (1.4.x) or better.

Category of the software product

OpenWFE contains the following features:
- Modelling, processing, and management of business workflow
- Worklist component for storing workitems (tasks) for participants,
- An APRE component, allowing you to implement automated agents into your work flows,

- Droflo web-based flow designer;a well-documented REST interface; and several libraries to access it.

Supported interfaces

Interface 1: OpenWFE uses his own Process definition language - POOL (Process Oriented OpenWFE Lisp/Language). There were some trials to build the xpdl2owfe tool, but the subproject was stopped. Currently it is not possible to exchange process model with other PDLs.

Interface 2: There are three implementations of worksession interface: **RMI, REST** and **SOAP**.
The REST interface can be accessed by:

- python application using the Pyya (python library)
- .NET applications using the library written in C# (can be compiled as a DLL)
- Perl application using the OpenWFE-Perl module

Interface 3: The administration tool is available through the web interface. The chosen user can monitor the workflow and participants' stores change the work items, delegates work items to other participants of the workflow. The user can also look at the history of the workflow process.

Additional interfaces:

By default, OpenWFE stores all data in files. It is possible to store whole xml files in database or LDAP.
There are currently 2 implementations of a Workitem Store, one that keeps items in files and one that store them in a relational database system (MySQL and PostgreSQL for instance, or any SQL-92 compliant DB). The Participant Map is by default an XML file, but it can easily be implemented as an LDAP directory.

4.15. Oracle BPEL Process Manager

Introduction

Oracle BPEL Process Manager implements the BPEL4WS specification version 1.1 [4.14] together with a series of extensions that are partially related to the upcoming BPEL specification version 2.0 [4.15]. As depicted in Figure 4.20 [4.16] it consists of four building blocks:

- *BPEL Designer*: a tool supporting to model, edit, design, and deploy BPEL processes. BPEL Designer allows viewing and modifying the BPEL source code. It is available with Oracle JDeveloper or as a plug-in on the Eclipse Platform version 3.0GA.
- *Oracle BPEL Server*: the server, to which BPEL process will be deployed. It contains human workflow, technology adapters, and notification services components.
- *Oracle BPEL Console*: a console to run, manage, and test deployed BPEL process. It provides a Web-based interface for management, administration, and debugging of processes deployed to the Oracle BPEL Server.
- *Oracle Database Lite*: a lightweight database that holds the BPEL schema (Windows only).

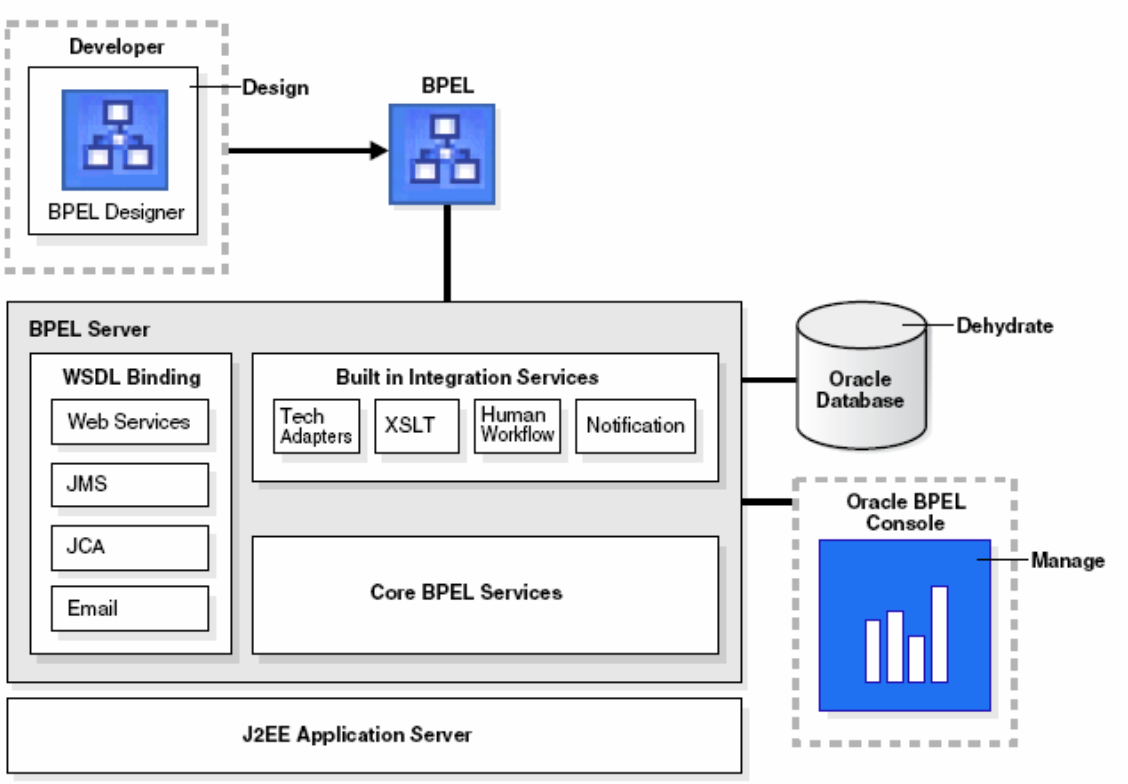


Figure 4.20 - Components of the Oracle BPEL Process Manager

Description

Oracle BPEL Designer and the *Oracle BPEL Process Manager* for JBoss are based on the Collaxa tools (designer and BPEL engine) that have been bought by Oracle in 2004. Both tools have been used by FhI FOKUS in student projects at the Technical University Berlin as well as in international research projects. Thus, their evaluation is based on practical experience.

The extensions to BPEL4WS 1.1 introduced by Oracle are supported by the *Oracle BPEL Process Manager for Developers*. This tool has been evaluated especially in the context of VISP. The functionality of *Oracle BPEL Process Manager for Oracle AS Middle Tier* has been considered on the bases of the accompanying documentation.

- Oracle BPEL Designer (evaluated)
- Oracle BPEL Process Manager for JBoss (evaluated) or WebLogic application server
- Oracle BPEL Process Manager for Developers (evaluated)
  - JDeveloper BPEL Designer
  - Oracle BPEL Server
  - Oracle AS J2EE Container
  - Oracle Database Lite
- Oracle BPEL Process Manager for Oracle AS Middle Tier
  - Oracle BPEL Server
  - Runtime services
  - Adapters
  - Requires Oracle AS 10g
  - Oracle Database Lite

Category

BPEL Designer is a graphical and textual BPEL editor. It uses WSDL 1.1 [4.17] specifications of the invoked services as input data and produces a BPEL 1.1 specification together with the provided WSDL 1.1 specification as output data.

After validating the BPEL specification, the designer compiles it into a Java .jar file that can be deployed on the supported application servers.

BPEL Server is a workflow execution tool. It runs within a Java application server like JBoss, WebLogic, or Oracle AS. The process manager comprises a SOAP engine to communicate with other services or to be invoked by its clients. It can be controlled and configured by the *Oracle BPEL Console*, a Web based management tool.

For the support of long lasting workflows, the process manager uses a lightweight respectively a complete database to "dehydrate" and store the workflow instances.

Interfaces

Interface 1: *BPEL Designer* can be used to create a BPEL specification from scratch or to import, modify, and extend existing BPEL choreographies. The first case is guided by tool specific wizards. In the second case, the appropriate configuration files have to be created manually. At least the XML configuration file that maps partner links to WSDL files and the WSDL file containing the description of the provided service have to be defined.

Interface 2: *BPEL Designer* creates a WSDL file that contains the service definition of the BPEL workflow including the corresponding partner link types.

Interface 3: *BPEL Server* is able to invoke Web services that provide a valid partner link type in the associated WSDL. This includes BPEL processes running in the *Oracle BPEL Server*. The WSDL can be retrieved using UDDI and WS-Inspection [4.22] APIs used by a *BPEL Designer* wizard. Thus, the designer is able to browse WSDL repositories to retrieve the needed WSDL definitions.
Between BPEL processes the following interaction patterns are explicitly supported and explained in [4.19]

- One-Way Message
- Synchronous Interaction
- Asynchronous Interaction
- Asynchronous Interaction with Timeout
- Asynchronous Interaction with a Notification Timer
- One Request, Multiple Responses
- One Request, One of Two Possible Responses
- One Request, a Mandatory Response, and an Optional Response
- Partial Processing
- Third-Party Interactions

It has to be noticed that *BPEL Server* uses WS Addressing to identify BPEL process and Web service instances. If an invoked service is stateful or third party interactions have to be supported the invoked Web services either have to support WS-Addressing, too, or BPEL correlation sets have to be introduced.

Additionally *Oracle BPEL Server* supports so-called *workflow services* that allow including human users in specified roles in the BPEL workflows. Such a feature is currently part of the ongoing BPEL 2.0 standardisation.

Interface 4: The interaction patterns between different BPEL processes have been introduced above. Cooperation with other workflow execution platforms seems to be possible as long as they satisfy the WSDL/SOAP conventions together with WS-Addressing that are necessary to collaborate with *BPEL Server.*

Interface 5: *BPEL Console* allows configuring, monitoring, debugging, and controlling BPEL processes and instances. *BPEL Console* is a Web based application that communicates with *BPEL Server* using proprietary interfaces.

Auxiliary interfaces: *BPEL Server* offers technology adapters [4.23] to access various resources via generated WSDL interfaces. This feature has not been evaluated, yet.

- File adapter
- FTP adapter
- Advanced queuing service adapter
- Database adapter
- JMS adapter
- Oracle application adapter

BPEL Designer includes a UDDI browser to retrieve information about available WSDL specifications.

Supported standards

The software supports the **BPEL 1.1** standard, in detail it supports the activities:

- assign
- invoke
- reply
- receive
- wait
- terminate
- throw
- compensate
- empty

Together with the control elements

- scope
- switch
- while
- pick
- flow

Additionally it supports **BPEL 2** and other enhanced features like

- Java embedding
- XPath extension functions
- Support of *BPEL Process Manager Services* like:
    - XSLT mapper and transformations

- Workflow services
- Worklist applications
- Notification services (eMail, voise, fax, SMS, pager)
- Sensors

The latter enhancements are only supported by the *JDeveloper BPEL Designer*. Consequently, incomplete, abstract BPEL 1.1 and WSDL 1.1 specifications can be imported into the *BPEL Designer* that produces complete, executable BPEL 1.1 and WSDL 1.1 specifications. *BPEL Designer* uses the concept of *partner link types* as defined in [4.29] to link BPEL and WSDL specifications.

4.16 ITP-Commerce Process Modeler for Microsoft Visio

Introduction

Process Modeler is a Microsoft Visio plug-in. It enables to design business processes in Microsoft Visio utilizing BPMN 1.0. It fully supports the BPMN 1.0 standard, including all BPMN elements and all BPMN attributes defined by specification. Visio stencil consists of the 15 BPMN basic elements:

- Start event
- End event
- Intermediate Event
- Task
- Sub-Process (collapsed)
- Sub-Process (expanded)
- Gateway
- Sequence flow
- Message flow
- Assertion
- Text annotation
- Data object
- Group
- Pool
- Lane

The tool offers the following features:

- BPMN Diagram Validation
- User defined attribute maintenance using an Attribute Explorer
- Migration BPMN 0.9 to 1.0
- BPMN style export
- Element numbering
- Process documentation
- BPEL 1.1 mapping and export with special support including round trip engineering for *Oracle BPEL Process Manager*

- XPDL 1.0 mapping and export with special support including round trip engineering for the *Carnot process engine*[1]

ITpearls commerce started developing back in 2002 a modeler to run on every business analyst's desktop. *Process Modeler* 2.1 is a typical plug-in for Microsoft Visio that implements the BMPN specification 1.0 and that uses the functionality of the widespread Microsoft Office product family to generate an environment for designing BPNM diagrams. *Process Modeler 2.1* is available in English, German and Japanese. It comprises:

- BPMN graphical editor with validation checking. Because *Process Modeller* is a plug-in for Microsoft Visio, it is possible to built BPMN diagrams via drag and drop.
- Capability to enrich BPMN models with Web services instrumentation in order to export valid BPEL 1.1 and XPDL 1.0 definitions. Compliant BPEL infrastructures, like Oracle's *BPEL Process Modeller,* are able to import this process definition for final orchestration and deployment.
- Capability to map/export BPMN diagrams to XPDL 1.0. Compliant XPDL infrastructures like *Carnot's process engine* are able to import this process definition for final orchestration and deployment.

Because BPMN is no executable language, BPMN tools cannot support the interfaces identified in the WfMC reference model. Instead, a BPMN tool has interfaces to other workflow editors like BPEL or XPDL editors. Like most of these tools, *Process Modeler* can import WSDL specifications of existing services. The WSDL specifications can be used to specify BPMN interfaces and are part of the export to other workflow editors.

The tool can import

- WSDL: *Process Modeler* has to import WSDL files before it is possible to transform BPMN diagrams to BPEL specifications.
- *Process Modeler 2.1* allows converting diagrams witch had been created with *Process Modeler 1.1* based on BPMN specification 0.9 to the latest version.

The tool can export

- BPEL 1.1: BPMN diagrams are mapped to BPEL 1.1 with optional support for *Oracle BPEL Process Manager.*
- XPDL 1.0: BPMN diagrams are mapped to XPDL1.0. The exported files are ready to be used in XPDL 1.0 editors like *Carnot process engine*.
- BPMN diagrams as image files.

Category

Process Modeler is a modelling tool.

Interfaces

As a modelling tool, *Process Modeler* does not support any of the interfaces identified in the WfMC reference model. Instead, it exports abstract BPEL 1.1 and XPDL 1.0 specifications that can be refined utilizing a BPEL or XPDL modelling tool.

Recommendation

[1] *Carnot process engine* is an eclipse based XDPL editor and engine provided by Carnot AG http://www.carnot.ag/

If we are interested in Process Modeler for VISP, we should observe whether an adequate documentation would be released in future. The integration in VISIO and thus in the typical working environment of a business modeler is a strong advantage of the tool. The availability of mapping to favoured orchestration languages like BPEL and XPDL is another advantage of the tool.

4.17 SAP NetWeaver Exchange Infrastructure

Workflow software overview

SAP NetWeaver Exchange Infrastructure 3.0 is the latest version of the NetWeaver platform.

NetWeaver is an integration and application platform. SAP NetWeaver unifies integration technologies into a single platform and it is preintegrated with business applications, reducing the need for custom integration. The platform is based on industry standards and can be extended with commonly used development tools such as Java 2 Platform, Enterprise Edition (J2EE); Microsoft .NET; and IBM WebSphere, Figure 4.21.

Workflow software description

SAP NetWeaver Exchange Infrastructure (SAP XI) provides open integration technologies that support process-centric collaboration among SAP and non-SAP applications, both within and beyond enterprise boundaries.
It provides many adapters such as SOAP, JMS, JDBC to interconnect with heterogeneous systems, Figure 4.22.

Category of the software product
Process modeler and engine

Supported interfaces

Interface 1: Possibility to import BPML and BPEL definitions.

Interface 2: Workflow clients can be of various types' grace of the great number of adapters available to communicate with external applications (see Interface 3).

Interface 3: SAP XI provides web services support and provides a set of adapters. Remote applications can be hosted on SAP systems (via IDOC/RFC), JAVA applications with JCA. It also supports JMS, Corba, Tibco, HTTP(S), FTP, and SMTP. Application adapters: Ariba, BEA WLI, Lotus Notes, Microsoft CRM, oracle Applications, PeopleSoft, BEA Tuxedo…
It also supports the following industry specific B2B adapters: UCCnet, SWIFT, and RosettaNet.
Interface 4: Workflow collaboration can be achieved with web services ends.

Interface 5: NetWeaver provides a "Central monitoring" feature: messages, components, performance can be monitored and alert messages can be sent.

Additional interfaces: Netweaver provides adapters for B2B like RosettaNet, UCCNet, CIDX.

Supported standards

Workflow execution tools: NetWeaver XI runs BPEL process on its execution engine (Integration Server), Fifure 4.22.

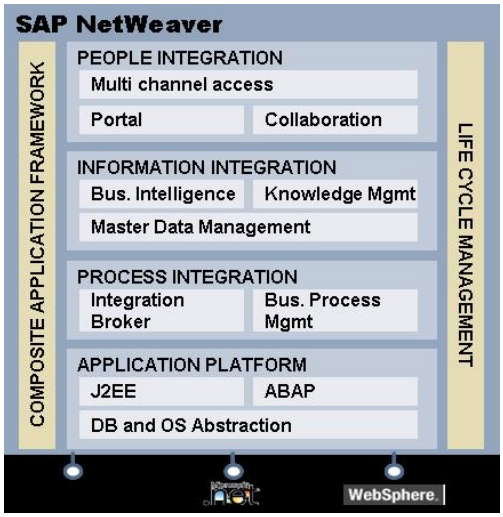


Figure 4.21. SAP NetWeaver Exchange structure

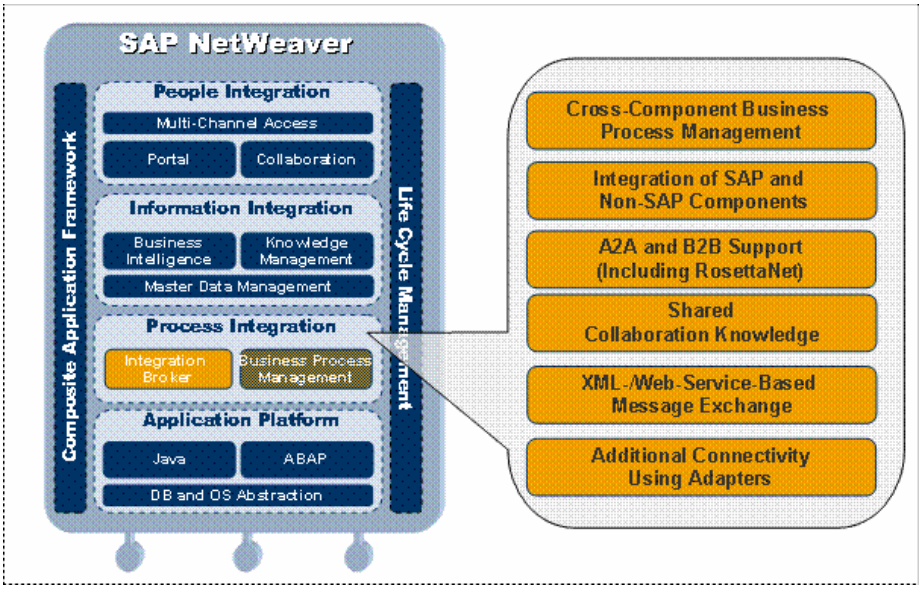


Figure 4.22. SAP NetWeaver Exchange structure

4.18 WfMOpen

Workflow software overview

WfMOpen is a J2EE based implementation of a workflow facility (workflow engine) as proposed by the Workflow Management Coalition (WfMC) and the Object Management Group (OMG).

WfMOpen is based on the WfMCore project that started in summer 2001 as a development initiative within Danet GmbH. It was triggered by the observation that implementing configurable workflow processes was a recurring task in several software development projects.

Workflow software description

The workflow component includes and it is based on a set of JAVA interfaces that define an API for a workflow management facility. These "omgcore" interfaces follow OMG's *Workflow Management Facility Specification, V1.2* very closely, while making some changes to adapt the CORBA service to the established design practices for a Java API.
The workflow engine is designed as a J2EE component. It is intended to be integrated in an application that requires a workflow engine.

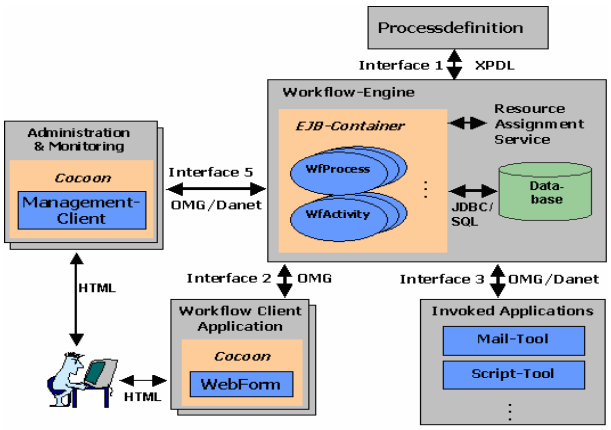


Figure 4.23 – The workflow engine's architecture

The previous image gives a view of engine architecture. The central component is the workflow engine. It consists of a set of EJBs and some JMS queues packaged as an EJB jar that can be integrated in any application. The workflow engine can be used without any of the other components of the project provided that all management functions (e.g. importing process definitions) are implemented by your client(s).

Moreover, the engine offers a set of tools like:

- JavaScript interpreter (based on Rhino JavaScript interpreter)
- Jelly (a tool for turning XML into executable code)
- LDAP queries and manipulation
- A SOAP tool which invokes web services dynamically
- XSLT
- Mail (javax mailing service)
- Wait (a tool can be used for process delay)
- Channel based Access (Tools that support message oriented communication with process instances)
- Tool Agents (the possibility to add your „tool-plugins" to the workflow engine)

Category of the software product

The WfMOPEN engine is an Open Source implementation of an XPDL engine written in Java. It offers a set of java libraries (API), DBMS connectors and tools to support workflow environments.

Supported interfaces

Interface 1: Exchange of workflow descriptions (models, choreographies, orchestration) between the tool and other modelling tools, repositories, workflow execution tools.
The tool can import and export to XPDL 1.1 and XML format using WfMC specifications.

Interface 2: Supports the invocation of workflow engines by workflow clients; Implementation of the interface by WSDL interface or a programming language interface.
Clients invoke engine, using java API.
The engine client API follows the OMG's *Workflow Management Facility Specification*.

Interface 3: Application invocation, which perform additional workflow activities; Implementation of the interface (by WSDL interface, programming language interface; notification mechanism for human actions).
The WfMOPEN engine can invoke external applications, which are involved into workflow process. In the engine architecture figure (see previous paragraph), it is clear that applications are called using OMG asynchronous mechanisms and those are supported by engine tools.

Interface 4: Support of workflow integration and collaboration (between different domains and/or distributed workflow execution platforms.
There is no specific support. Integration must be realized ad hoc.

Interface 5: Monitoring and administration of the workflows activities; Life cycle management of distributed workflows.
Monitoring and administration are supported. In fact, as well as a Java API, the engine comes packaged with a simple web application (which supports user authentication mechanisms).

By this application, an administrator can manage:

- Process Definitions: lists all currently imported process definitions and allows the import of additional process definition files.
- Processes: details of a selected process can be viewed and management operations can be performed.
- Assigned Activities: it lists all known resources for a detail view of the selected resource's activity assignments.
- Staff Members: lists all members of the staff known to the resource assignment system. Staff members may be added, edited or removed.
- Groups: lists all staff groups known to the resource assignment system. Groups may be added, edited or removed.

Additional interfaces: import/export of data towards other products like databases, file systems, repositories (UDDI, LDAP); monitoring of transactions.
WfmOPEN supports explicitly XPDL, Wf-XML, SOAP, ASAP, LDAP and even for several DBMS. Moreover, it can be extended by Java modules in order to embrace other standards.

Supported standards

The WfmOpen engine supports projects based on WfMC specifications expressed by XPDL format.

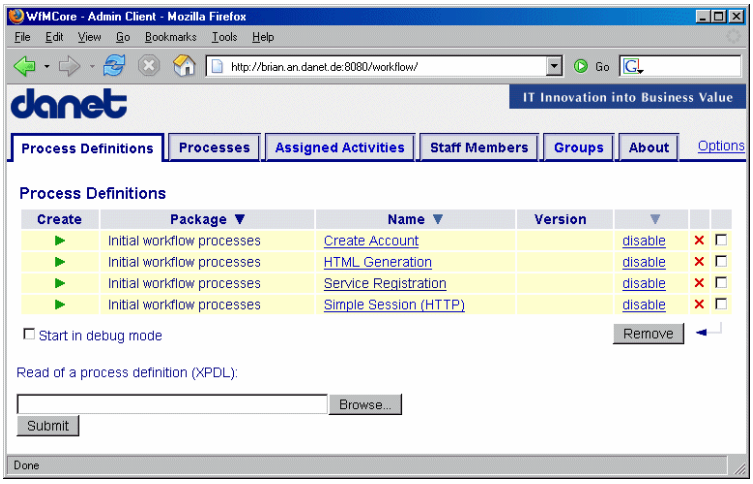


Figure 4.24 – The workflow engine's application

4.19 YAWL

Workflow software overview

YAWL is a new, powerful workflow language (Yet another Workflow Language). It was inspired by the Petri nets as a starting point and mechanisms to allow for a more direct and intuitive support of the workflow patterns have been added.

While not a commercial language itself it encompasses these languages, and, in addition, it has a formal semantics.

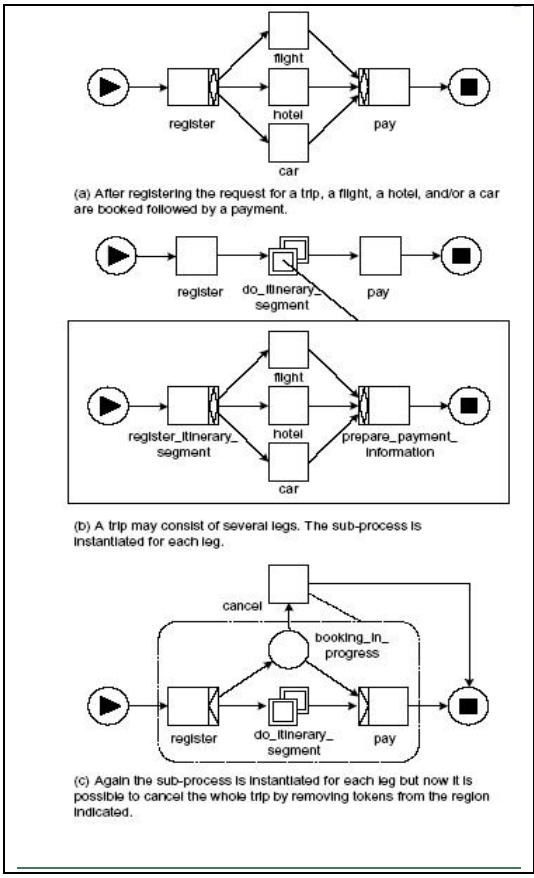
YAWL supports the control-flow perspective and the data perspective. However, it is important that relations with the resource perspective are investigated and formalized.

YAWL language supports the following elements: atomic tasks, composite tasks, conditions, one input condition and one stop condition. Each task can have multiple instances. A task can be AND/OR/XOR-split task or AND/OR/XOR-join task. YAWL provides a notation for removing tokens from one specified region – the cancellation region. Figure 4.25 presents these elements in detail.

Workflow software description

YAWL environment is composed of:

- the YAWL specifications (Figure 4.25); YAWL Editor (Figure 4.27) and YAWL Editor Lite versions; YAWL Engine (Web application) (Figure 4.26); YAWL Standalone (a Java standalone application with the same functionalities as the Web YAWL Engine)



Definition of task **register**

```
<task id="register">
<name>Collect information from customer</name>
<flowsInto>
<nextElementRef id="flight"/>
 <predicate>/data/want_flight = 'true'</predicate>
 <isDefaultFlow/>
</flowsInto>
<flowsInto>
<nextElementRef id="hotel"/>
<predicate>/data/want_hotel = 'true'</predicate>
</flowsInto>
<flowsInto>
<nextElementRef id="car"/>
 <predicate>/data/want_car = 'true'</predicate>
</flowsInto>
<join code="and"/>
 <split code="or"/>
<startingMappings>
 <mapping>
 <expression query="/data/customer"/>
 <mapsTo>customer</mapsTo>
 </mapping>
</startingMappings>
<completedMappings>
<mapping>
 <expression query="/data/customer"/>
 <mapsTo>customer</mapsTo>
</mapping>
<mapping>
 <expression query="/data/want_flight"/>
 <mapsTo>want_flight</mapsTo>
</mapping>
....
</completedMappings>
<decomposesTo id="register"/>
</task>
```

Figure 4.25 YAWL Specifications

Category of the software product

Workflow specifications in YAWL language are designed using the YAWL Editor and deployed into the YAWL Engine, which stores these specifications in the YAWL Repository. The YAWL Repository manages a collection of "runable" workflow specifications. After deployment, workflow specifications can be instantiated through the YAWL engine, leading to workflow instances (or *cases*). The engine handles the execution of these cases. The engine can be invoked using the Web interface, or from client WS (Web Services) via YAWL WS Invoker interface.

Supported interfaces

The YAWL engine supports all the 5 interfaces of the WfMC reference model: (A) interfaces capturing the interactions between the YAWL designer and the YAWL manager on the one hand, and the YAWL engine on the other; and (B) interfaces capturing the interactions between the YAWL services and the YAWL engine. The group of interfaces (A) corresponds to **Interface 1** (Process Definition tools) and **Interface 5** (Administration and Monitoring tools). The group set of interfaces (B) corresponds to WfMC's **Interface 2-3** (Workflow Client Applications and Invoked Applications), and **Interface 4** (Workflow Interoperability). Both interfaces (A and B) are specified in WSDL. Users interact with the YAWL system through a Web browser. Notice that, for **Interface 1** YAWL workflows are stored in a XML file, which does not comply with any standard. Neither the engine nor the designer does not offer support for import/export facilities from/to other WF languages.

Additional interfaces

Import/export of data to other products (databases, repositories, etc.) is not supported.

Supported standards

a) The process is modeled using YAWL Editor (GUI implemented in Java Swing). YAWL is an advanced workflow language. Complex workflows can be modeled easily. Two mechanisms for linking one process to another are provided:

- static linking (design time) directly using YAWL editor and language constructs;
- Dynamic linking using YAWL worklets (see worklet description below). The selection of the invoked worklet is done at runtime, based on a *selection condition*.

YAWL Engine has a Web-based interface (HTML). There is also a standalone version of the engine, written in Java Swing for testing and evaluating purposes.

b) Workflow execution tools: YAWL Engine has been implemented in Java using XML-based standards such as XPath, XQuery, and XML Schema. The technical environment, needed for the YAWL Engine, includes:

- Java 2, Standard Edition (J2SE), 1.4.2_07; Apache Tomcat 5.0.28; PostgreSQL – 8 (if you wish to use database persistence).

YAWL Editor needs Java 2 Standard Edition. Additional libraries are packed in a .jar file. No additional software is needed in order to run the editor.

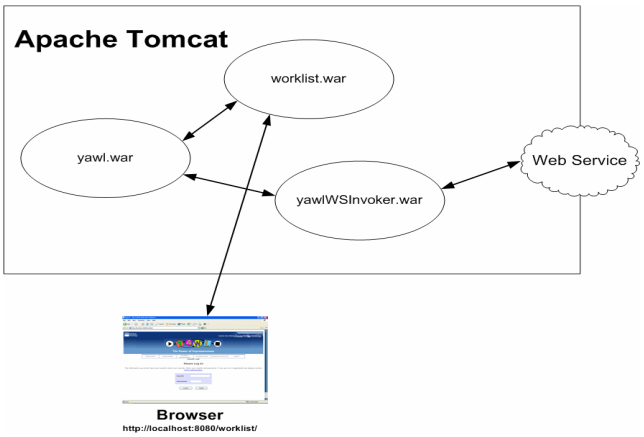


Figure 4.26 YAWL Engine

YAWL Engine Architecture

The environment of a YAWL system is composed of *YAWL services*. End-users, applications, and organizations are abstracted as services in YAWL. YAWL services include: (1) *YAWL worklist handler*, (2) *YAWL web services broker*, (3) *YAWL interoperability broker*, and (4) *custom YAWL services*. The YAWL worklist handler corresponds to the classical worklist handler (the *inbox*) and it is used to assign work to users of the system. Through the worklist handler, users can accept work items and signal their completion. In YAWL, the worklist handler is considered a service decoupled from the engine. The YAWL web services broker is the glue between the engine and other web services. It is unlikely that web services will be able directly connect to the YAWL engine. It is desirable not to adapt the interface of the

engine to suit specific services. The YAWL web services broker acts as a mediator between the YAWL engine and external web services. The YAWL interoperability broker is a service designed to interconnect different workflow engines. To illustrate that there is not a fixed set of YAWL services we included a custom YAWL service. A custom service connects the engine with an entity in the environment of the system.

YAWL Editor

Figure 4.27 is a screen-shot of YAWL Editor version 1.3.02. The Editor incorporates and/or uses components of the following tools: YAWL Engine, JGraph 3.1, Proguard 3.2, WofYAWL 0.4 and PostgreSQL 8.0. The Editor offers the facility to validate the workflow, export the workflow to YAWL engine and define workflow variables.

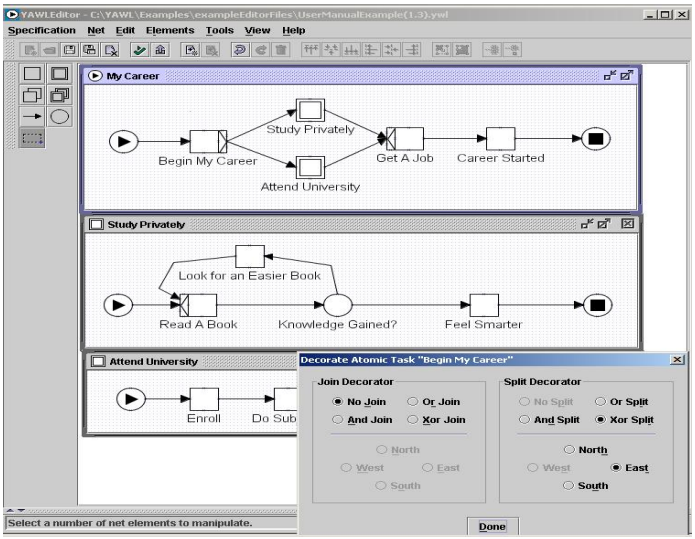


Figure 4.27 - YAWL Editor

What is the YAWL Worklet Service?

The *Worklet Dynamic Process Selection Service* for YAWL provides the ability to substitute a workitem in a YAWL process with a dynamically selected "worklet" - a discrete YAWL process that acts as a sub-net for the workitem and so handles one specific task in a larger, composite process activity. An extensible repertoire (or catalogue) of worklets is maintained. Each time the service is invoked for a workitem, a choice is made from the repertoire based on the data within the workitem, using a set of rules to determine the most appropriate substitution. The workitem is checked out of the YAWL engine, and then the selected worklet is launched as a separate case. The data inputs of the original workitem are mapped to the inputs of the worklet. When the worklet has completed, its output data is mapped back to the original workitem, which is then checked back into the engine, allowing the original process to continue. Worklets can be substituted for atomic tasks and multiple-instance atomic tasks. In the case of multiple-instance tasks, a worklet is launched for each child workitem. Because each child workitem may contain different data, the worklets that substitute for them are individually selected, and so may all be different. The repertoire of worklets can be added to at any time, as can the rules base used for the selection process. Thus the service provides for dynamic ad-hoc change and process evolution, without having

to resort to off-system intervention and/or system downtime, or modification of the original process specification.

4.20. Survey of UML Modelling Tools

Deliverable D2.1 has identified the Unified Modelling Language UML 2 as a possible candidate to be utilized for the specification of choreographies. The specific modelling element that should be used for the definition of choreographies is *UML activity diagrams*. Because almost every UML tool supports the definition of such diagrams, the evaluation of UML tools as part of the work in task 2.2 seems not to be meaningful. Instead, we will reference to available evaluations of UML 2 tools. If the need to select such a tool will arise in VISP, the appropriate links have to be checked respectively the partners' experience with such tools has to be considered.

- A comprehensive list of available UML tools is provided at http://www.objectsbydesign.com/tools/umltools_byCompany.html.
- An overview and comparison of selected properties of more than 30 UML tools can be found at http://www.oose.de/umltools.de.
- An actual evaluation of ten UML 2 tools from November 2005 in German can be found in [Oestereich *et al.,* 2005]

This evaluation compares commercial UML products including *Borland Together Architect* that has been evaluated in VISP as a BPMN modelling tool:

- ARTiSAN Studio 6.0
- Borlang Together Architect 2006-02-10
- Poseidon for UML 3.2 (community edition)
- IBM Rational Software Architect 6.0
- IBM Rational Software Modeler
- ARIS UML Designer 7.0
- objectIF 5.0
- innovatorAOX 2006-02-10 EclipseUML 2.1
- Enterprise Architect 5.0
- Telelogic TAU 2.5

The evaluation considers the following criteria:

- General information
- Supported platforms / requirements
- Supported interfaces
- Supported UML diagram types
- UML2 support
- Code generation
- Reverse engineering
- Support for MDA
- Support for XMI
- Generation of documentation

4.21. Assessed Software and Corresponding Covered Technologies

The following table presents the short-list workflow software, assessed by the partners, and the corresponding workflow technologies and standards, which these tools are based on.

Table 4.7 – The assessed short-list workflow software and corresponding workflow technologies

№	Assessed Products	Covered Technologies
1	ActiveBPEL Engine (Metaware)	XML, BPEL, WSDL, SOAP, XPath, XQuery
2	Active Endpoints ActiveWebflow Standard (ICCS)	XML, BPEL, WSDL, SOAP, XPath, XQuery
3	ActiveWebflow Designer (ICCS)	XML, BPEL, WSDL
4	Biztalk Server (VALTECH)	XML, BPEL, WSDL, SOAP, XPath,EDI, RosettaNet
5	Borland Together Designer 2006 (FhI Fokus)	BPMN, UML
6	Cape Clear (VALTECH)	XML, BPEL, WSDL, SOAP
7	con-cern (WUT)	SOAP
8	Enhydra JaWE (ICCS)	XPDL, Wf-XML
9	Enhydra Shark (ICCS, Metaware)	XPDL, Wf-XML, SOAP, ASAP
10	enterpriseXtention (UAM)	
11	Freefluo (Metaware)	WSFL, WSDL
12	JBoss jBPM (VALTECH)	LDAP, jPDL
13	ObjectWeb BONITA (ICCS)	XPDL, SOAP, WSDL
14	OpenWFE (UAM)	XML
15	Oracle BPEL Process Manager (FhI Fokus)	XML,BPEL, WSDL, SOAP, UDDI, XPath
16	Process Modeler (FhI Fokus)	BPMN
17	SAP NetWeaver Exchange Infrastructure (VALTECH)	XML, BPEL, WSDL, SOAP,UDDI, WS Security
18	WfMOpen (Metaware)	XPDL, Wf-XML
19	YAWL (WUT)	WSDL

5. Software Evaluation Methodologies

In Chapter 3 it has been performed an overview of different methodologies, applied for the evaluation of the software products. The goal of this chapter is to motivate a common methodology, which can be applied for the comparison of relatively different workflow software products. Particularly the interested set of products for the VISP project are those software suits, which concern the modelling, specification and execution of activities, related to workflow management processes. Despite the particular domain of application, these software products respect common requirements, needed from various program products. Taking into account the overview in Chapter 3 of the available and used approaches for evaluation of software products this development is targeting a minimization of the subjective influences, which the experts can introduce by choosing their own evaluation scheme and by grouping a corresponding set of criteria. Taking into account the overview of the recommended criteria for software evaluation, this work tries to find a common basis to assess the prospective products, which can be used for the developments in the VISP project.

The overview in chapter 3 demonstrates that there are several approaches in applying methodologies for assessment of software products (3.3.1 – 3.3.5). Particular attention is done for those methodologies, which address information systems, and products, which commonly operate in global network environment and Internet surroundings. This attention is influenced by the requirements of the VISP project the software tools to serve and support Internet functionalities in message generation and data processing, which are natively used in workflow management systems. The evaluation scheme, which is targeted in this chapter, has to minimize the subjective influence of the evaluation, performed from different experts to different software products.

The experience, which exists until now about the particular domain for methodology development for assessment of software and informational products, is not mature to propose to users a common easy and useful methodology for products evaluation. The evaluation methodologies does not found on an objective background, which can minimize the subjective expert influence. As an attempt to find, a common objective background for the evaluation of software product this deliverable introduces the standardization approach in the assessment of the quality of the software products ISO/IEC 9126. Thus, a common objective methodology is used for the particular problem in product evaluation for this D2.2 deliverable. For the next section, common notions of the standard ISO/IEC 9126 are introduced by means to give explanations to the particular criteria, which are used as their hierarchical subordination for the evaluation of the quality of the software products.

5.1. ISO/IEC 9126 Evaluation Plan

To minimize the subjective influence of the evaluators for the assessment process of software products, it is beneficial to base the evaluation scheme on common well-accepted methodology. Our choice was to find appropriate standardization approach, which gives ground of this work to be performed in independent way from the applied assessment scheme. As successful solution for objective evaluation methodology in this deliverable, it has been chosen the international standard ISO/IEC 9126. It specifies a common framework

for evaluation and comparisons of software products. The standard ISO/IEC 9126 works out a quality model that categorises software quality attributes into six characteristics, which are further sub-divided into sub-characteristics [5.1], [5.2].

These sub-characteristics are used externally when the software is used as a part of a computer system, and their assessment results define the product quality, which is an inexplicit influence by the internal software attributes. The software quality is assessed, addressing the user needs, by the definition of evaluation criteria of the product quality in use. The standard ISO/IEC 9126 describes software quality metrics based on internal software attributes and external computer system behaviour. These types of metrics are applicable when specifying quality requirements and design goals for software products, including intermediate products. The characteristics and sub-characteristics provide consistent evaluation methodology for the software quality. They also provide a framework for specifying quality requirements for software and making trade-offs between different software product capabilities such as functionality, reliability, usability and efficiency.

The standard ISO/IEC 9126 enables the software product quality to be specified and evaluated from different perspectives by those associated with acquisition requirements, development, use, evaluation, support, maintenance, quality assurance, audit of the software.

The quality model defined in the standard ISO/IEC 9126 can be used to:
- validate the completeness of the requirements definition;
- identify software requirements;
- identify software design objectives;
- identify software testing objectives;
- Identify user acceptance criteria for a completed software product.

Any specification for the evaluation of the software quality conforming to ISO/IEC 9126 results from an evaluation process that considers the potential relevance of the product to:
- criteria for internal metrics of the product quality;
- criteria for external metrics of the product quality
- Criteria for quality in use metrics.

Product quality and the life-cycle

During the life cycle, the software endures much quality. The product quality at the beginning of the life cycle differs from the actual or to the delivered product quality. The assessment of the quality is also a reflection of different points of view of the evaluator. It is necessary to define in advance the set of evaluation criteria about the quality of the product and to follow the changes in this quality along the life cycle of the product. The standard considers several categories of the product quality for the different stages during the life cycle of the product. These quality categories are shortly introduced below. Practical implementation for the D2.2 works has the quality category **Quality in Use** (QIU).

Goal Quality (GQ*)* is a necessary and sufficient quality that reflects the real user needs. ISO 8402 defines the **quality** in terms of the ability to satisfy stated and implied user needs. Needs defined by a user do not always reflect the real user needs, because a usually the user is not always aware with his real needs. In addition, these needs are not constant and they may change in time. GQ is a conceptual entity, which cannot be completely defined at the beginning of the product design, but the developers have to try to get closer to it. GQ does not

necessarily mean perfect quality, but necessary and sufficient quality. Requirements for GQ can be evaluated by measuring quality in use (QIU) when the product is complete.

Required Product Quality (RPQ) is what is actually defined in the quality requirement specification. Quality requirements should be defined in the quality requirement specification. RPQ should consist of not only the optimal requirements to the product, but also the minimum requirements, which it has to satisfy.

Design Quality (DQ) is the quality supported by the core parts of the product, achieved during the stage of it software design. This quality address the software architecture of the product, it program structure, the user interface design's strategy. DQ is the reflection of the design philosophy and implementation strategy. Details of the software quality may be improved during the tasks of code implementation and code testing.

Estimated Product Quality (EPQ) is the quality that is estimated or predicted for the final software product quality after the completition of every stage of the product development. This quality category is closely related to DQ.

Delivered Product Quality (DPQ) is the quality of the delivered product, typically tested in a simulated environment with simulated data. During the tests, the bigger part of the faults are discovered and eliminated. However, some faults may remain after testing.

Quality in Use (QIU) is the user's view of the quality of a system containing software, and is measured in terms of the result of using the software; "Users" refers to both end users and maintainers. The quality in the users' environment may be different from that in the developers' environment, because some functions may not be visible to a user, or may not be used by a user. The user evaluates only those characteristics of software, which are visible to him during the processes of product utilization. Sometimes, software characteristics specified by an end user during the requirements analysis phase, no longer meet the user requirements when the product is in use, because of changing the user requirements and the difficulty of specifying the project needs.

The changes in the quality categories during the product life cycle are presented in Figure 5.1.

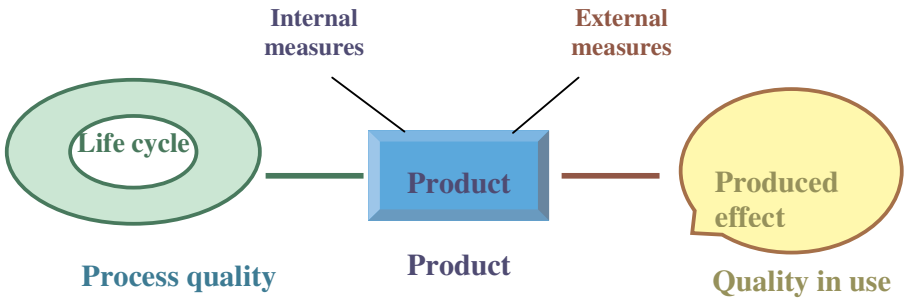


Figure 5.1 - Quality in the life-cycle

The evaluation of the products in order to assess the software product quality is the problem, which the standard tries to resolve, during the different stages of the product design, development, exploitation, maintenance. The software product quality can be measured internally (generally by static metrics for the code performance), or externally (generally by measuring the behaviour of the software). The design process quality contributes for the improvement of the product quality; respectively the product quality contributes for the improvement of the quality in use. Therefore, assessing and improving the quality of the design process is a means to improve the product quality, respectively in this chain by improving the product quality improvement of the quality in use is achieved. Hence, the requirements for having integral software product quality will influence the set of criteria for internal product quality, the criteria for external quality (as criteria for quality in use) by users.

Items to be evaluated

Items for quality assessments can be found by direct measurement, or indirectly by measuring their consequences and related influences. The software product quality may be estimated by measuring the performance of the user work, when the product is applied. The software product is assessed towards predefined set of external metrics which describe the product interaction with the working environment and which assesses the software in his operation functionality. The external quality that demonstrates the extent, to which the product satisfies the stated and implied user needs, can be measured in the operational environment by evaluating quality in use to achieve specified goals with effectiveness, productivity and user satisfaction. At the earliest stages of product life cycle, only resources and process can be measured. When intermediate products, specifications, source code, etc. become available, these can be evaluated by chosen internal metrics that can be used to predict values of the external metrics.

A distinction has to be made between the evaluation of a software quality of the product and the evaluation of the system quality in which the product is executed. The reliability of a system is assessed by observing all failures due to hardware, software, human error, etc. The reliability of the software product is assessed only by the failures due to faults, originating from product requirements, design or implementation mistakes in the software.

STANDARDISED Quality model

The software quality has to be evaluated according to a developed and defined quality model. The quality model has to be developed when quality goals for the software products are stated on the stage of product development. The international standard ISO/IEC 9126 provides and recommends a quality model, which can be used as a checklist of issues related to the quality, Figure 5.2. The attributes of the software product quality consist of a set of several main quality characteristics; each of them is constituted by a set of sub characteristics.

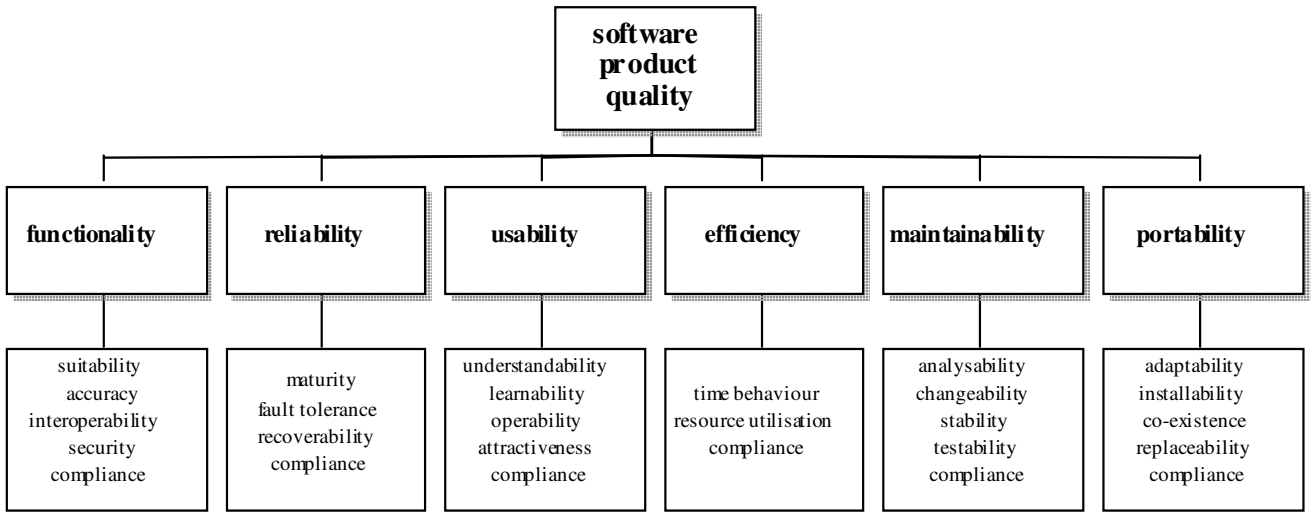


Figure 5.2 - Software product quality

The attributes of the software quality are categorised into six characteristics

- functionality
- reliability
- usability
- efficiency
- maintainability
- portability

The characteristics are sub-divided into sub-characteristics, which can be measured either by internal metrics or by external metrics.

The quality category "Quality in use" normally represents the user's point of view about the quality of the software product. To achieve appropriate "quality in use" means to meet quality criteria about external assessments of the product implementation. However, because the "quality in use" is a resulting quality characteristic from the external and internal quality of the product, the total product quality has to be estimated having assessment for the all quality categories of the product. Quality estimations are normally required for the all three levels, of the product development. However, satisfying the quality requirements for internal product measures is not usually sufficient to ensure appropriate quality for the product towards the criteria of external measures. Respectively satisfying the appropriate criteria for external measures does not guarantee quality in use, Figure 5.3.

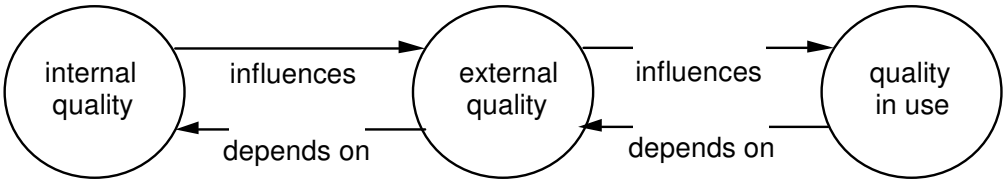


Figure 5.3 - Relationship between different types of quality

It is not practically possible to estimate all qualitative characteristics internally or externally for all components of a complex software product. The same case is met when many user tasks scenarios have to be assessed by means to define the appropriate quality in use. During the product quality assessment, it is necessary to allocate and to spend different resources, time, and cost for the product evaluation. The amount of the resources firmly depends from the business objectives, the application performance of the software, it nature and peculiarities of its design process.

Metrics for qualitative Attributes and characteristics

Terms and definitions
Quality

External quality: The extent to which the product satisfies a set of stated and implied needs when it is used under specified conditions.

Internal quality: The set of attributes of the product that determine the product ability to satisfy stated and implied needs when it is used under specified conditions.

Quality: The set of characteristics of a product that bear on its ability to satisfy stated and implied needs.

Quality in use: The extent to which a product can be used by specified users to meet their needs to achieve specified goals with effectiveness, task efficiency and satisfaction in a specified context of use.

Quality model: The set of characteristics and the relationships between them which provide the basis for specifying quality requirements and evaluating quality.
Software and user

Software: All or part of the programs, procedures, rules, and associated documentation of an information processing system. Software is an intellectual creation that is independent of the medium on which it is recorded.

Software product: The set of computer programs, procedures, and possibly associated documentation and data designated for delivery to a user. The set of the software products include also intermediate products intended for users like developers and maintainers.

User: An individual that uses the software product to perform a specific function. Users may include operators, recipients of the results of the software, or developers or maintainers of software.
Measurement

Attribute / Characteristic: A measurable physical or abstract property of an entity.

Direct measure: A measure of an attribute that does not depend upon a measure of any other attribute.

External measure: An indirect measure of a product derived from measures of the behaviour of the system of which it is a part.
The system includes any associated hardware, software (either custom software or off-the-shelf software) and users. The number of faults found during testing is an external measure of the number of faults in the program because the number of faults are counted during the operation of a computer system running the program to identify the faults in the code. External measures can be used to evaluate quality attributes closer to the ultimate objectives of the design.

Indicator: A measure that can be used to estimate or predict another measure.
The measure may be of the same or a different characteristic. Indicators may be used both to estimate software quality attributes and to estimate attributes of the production process. They are indirect measures of the attributes.

Measure (noun): The number or category assigned to an attribute of an entity by making a measurement.

Measure (verb): Make a measurement**.**

Measurement: The process of assigning a number or category to an entity to describe an attribute of that entity. "Category" is used to denote qualitative measures of attributes.

Metric: A measurement scale and the method used for measurement.

The levels of certain internal attributes have been found to influence the levels of some external measures, so that there is both an external aspect and an internal aspect to most characteristics. For example, reliability may be measured **externally** by observing the number of failures in a given period of execution time during a trial of the software, and **internally** by inspecting the detailed specifications and source code to assess the level of fault tolerance.

The correlation between internal attributes and external measures is never perfect and the effect that a given internal attribute has upon an associated external measure will be determined by experience, and will depend on the particular context in which the software is used. One can use direct and indirect measures for assessing software quality to obtain the estimations of the indicators. Some software attributes can be represented quantitatively by measuring software product parameters directly.

An indirect measure is derived from measures of one or more other attributes. A measure of an attribute from a running computer system is used as an indirect measure of attributes of the evaluated software, because the computer environment strongly influence the attributes of the software product.

Internal metrics Overview

Internal metrics can be applied to a non-executable software product, such as a program specification or source code, during the stages of design and coding of the software. The internal metrics measurements estimate the values of the internal attributes or indicate external attributes by analysing the static properties of the intermediate or final software products. The measurements of internal metrics use quantitative numbers about the frequency

how many software composition elements appear in the source code statements, the number of branches in the control graph, data flow and state transition representations.

External metrics Overview

External metrics apply quality metrics of the software product derived from measurements of the run-time behaviour of the system in which the software product is a part. The measurements are performed by testing, running and monitoring the executable software of the system.

The evaluation of the software quality of the product begins with the definition of static metrics, addressing issues of the product application area:

- business objectives related to the implementation of the software product;
- exploitation issues of the product;
- management of the product application in a specified organisational and technical environment

External metrics for the product quality is provided by users, evaluators, testers, and developers contributing to the "quality in use" evaluation if they are able to evaluate the software product quality during testing and operational tasks.

Relationship between external and internal metrics

When the software quality requirements are defined, the software quality characteristics or sub-characteristics, which contribute to the quality requirements, are classified. Then, the appropriate external metrics and acceptable levels of the metrics are specified to appropriate quantitative values. If the criteria achieves these values, this validate that the software meets the user requirements. Then the internal quality attributes of the software are specified, by means to achieve the required external quality characteristics. Appropriate internal metrics and acceptable levels are specified which quantify the internal quality characteristics that the software has to meet as internal quality during its development.

Quality in use metrics

"Quality in use" metrics assesses the extent to which a product meets the requirements of specified user group by means to achieve specified goals of **effectiveness, productivity and satisfaction** in an application context of the users. The assessment of the "quality in use" validates the software quality in specific user-task scenarios. "Quality in use" is the user's view of the quality for the system applying the software. The "quality in use" is measured in terms of the results when the software is used, rather than properties of the software itself. The"quality in use" represents the integral effect of the software quality in a practical user application. The relationship of "quality in use" criteria to the software quality characteristics depends on the type of user:

- "the quality in use" for the end user is mainly a result of functionality, reliability, usability and efficiency;
- "the quality in use" for a person, involved in the software maintenance is a result of maintainability;
- "The quality in use" for a person, engage in product deployment is a result of portability.

Quality in use is always influenced by any quality characteristics but it contains broader meaning than usability. The last addresses features, which concern only easy usage and attractiveness of the product.

Choice of metrics

The Workflow software tools can be evaluated from the following points of view:
- developer - using internal measures of the quality characteristics;
- person responsible for the implementation of the software in different environments - using metrics for portability;
- end user - using metrics for quality in use and external metrics for functionality, reliability, usability and efficiency;
- maintainer - using metrics for maintainability

Workflow software's quality characteristics and specifications

The quality model for the software evaluation of the Workflow software is chosen in this development to contain six characteristics:
- functionality
- reliability
- usability
- efficiency
- maintainability
- portability

For each quality characteristic several sub-characteristics are chosen, which contribute for the quality assessment of the software product.

Functionality

The functionality is the capability of the software to provide functions, which meet specified needs when the software is used under defined conditions. This characteristic is concerned with what the software does to fulfil needs, whereas the other characteristics are mainly concerned with when and how it does. The functionality sub-characteristics are the following:

Suitability - the capability of the software to provide an appropriate set of functions for specified tasks and user objectives.

Accuracy – it is the capability of the software to provide the right or agreed results or effects.

Interoperability – it is the capability of the software to interact with one or more specified systems.

Security - the capability of the software to protect information and data so that unauthorised persons or systems cannot read or modify them and authorised persons or systems are not denied access to them.
Compliance - the capability of the software to adhere to application related standards, conventions or regulations in laws and similar prescriptions.

Reliability

The reliability is the capability of the software to maintain its level of performance when used under specified conditions. The reliability sub-characteristics are the following:

Maturity – it is the capability of the software to avoid failure because of faults in the software.

Fault tolerance - it is the capability of the software to maintain a specified level of performance in cases of software faults or of infringement of its specified interface. The specified level of performance may include fail-safe capability.

Recoverability - the capability of the software to re-establish its level of performance and recover the data directly affected in the case of a failure. Following a failure, a software product will sometimes be down for a certain period of time, the length of which is assessed by its recoverability.

Compliance - it is the capability of the software to adhere to standards, conventions or regulations relating to reliability.

Usability

The usability is the capability of the software to be understood, learned, used and liked by the user, when used under specified conditions. Usability should address all of the different user environments that the software may affect, which may include preparation for usage and evaluation of results. The usability sub-characteristics are the following:

Understandability - it is the capability of the software product to enable the user to understand whether the software is suitable, and how it can be used for particular tasks and conditions of use. This will depend on the documentation and initial impressions given by the software.

Learnability - it is the capability of the software product to enable the user to learn its application.

Operability - it is the capability of the software product to enable the user to operate and control it.

Attractiveness - it is the capability of the software product to be liked by the user.

Compliance - the capability of the software to adhere to standards, conventions, style guides or regulations related to usability.

Efficiency

The efficiency is the capability of the software to provide the required performance, relative to the amount of resources used, under stated conditions. The efficiency sub-characteristics are the following:

Time behaviour - it is the capability of the software to provide appropriate response and processing times and throughput rates when performing its function, under stated conditions.

Resource utilisation - the capability of the software to use appropriate resources in an appropriate time when the software performs its function under stated conditions.

Compliance - it is the capability of the software to adhere to standards or conventions relating to the efficiency**.**

Maintainability
The maintainability is the capability of the software to be modified. Modifications may include corrections, improvements or adaptation of the software to changes in environment, and in requirements and functional specifications. The maintainability sub-characteristics are the following:

Analysability - it is the capability of the software product to be diagnosed for deficiencies or causes of failures in the software, or for the parts to be modified and identified.

Changeability - it is the capability of the software product to enable a specified modification to be implemented. Implementation includes coding, designing and documenting changes.

Stability - it is the capability of the software to minimise unexpected effects from modifications of the software.

Testability - it is the capability of the software product to enable modified software to be validated.

Compliance - it is the capability of the software to adhere to standards or conventions relating to maintainability.

Portability
The portability is the capability of software to be transferred from one environment to another. The environment may include organisational, hardware or software environment. The portability sub-characteristics are the following:

Adaptability - the capability of the software to be modified for different specified environments without applying actions or means other than those provided for this purpose for the software considered.

Installability - it is the capability of the software to be installed in a specified environment.

Co-existence - it is the capability of the software to co-exist with other independent software in a common environment sharing common resources.

Replaceability - it is the capability of the software to be used in place of other specified software in the environment of that software. Replaceability is used in place of **compatibility** in order to avoid possible ambiguity with interoperability. Replaceability does not imply that this software is able to replace the software under consideration.

Compliance - it is the capability of the software to adhere to standards or conventions relating to portability.

5.2 VISP Methodology for Comparison of the Workflow Software Products

Task 2.2 has to tackle the methodological difficulties for the evaluation and assessment of the workflow software products. The researches in D2.2 have to identify the products, which belong to the category "workflow management products". The overview, presented in Chapter 3, found a large number of products, which claim to belong to the categories, related to workflow management: modelling, specification, execution. During the work of WP2, a set of 134 software products has been identified. However, these products address different area of system applications (scientific systems, business systems). They are described according to the available references having different level of maturity, functionality, usability. Hence, to perform a practical investigation in depth the works in Task 2.2 had to be restricted to a short list of prospective products, which are accessible by the VISP consortium for testing.

However, the evaluation process has to tackle a methodological problem which origins from the fact that different experts assess different products. The qualification of the experts, their experience, the variety of the workflow software products, and the lack of common evaluation methodology – all these factors can strongly influence the results of the product evaluation.

A second methodological problem arises for the evaluation findings how to quantify the evaluation results to generate a common scale for products comparison. This scale is necessary to support the decision making process for finding a restricted set of prospective products for implementation in VISP project and even to make a unique choice which is the target objective for the designers of workflow management systems.

During the test processes, the VISP consortium tried to design its own methodology for evaluation and comparison of the software products. This methodology targets the minimization of the subjective influences coming from the different experts. The theoretical background of the VISP methodology is founded on consideration to minimize the noise influences, which take place in a control and management systems. The formal considerations are given below. The following notations are used:

A_i – assessment rate of software product i, $i=1,N$;

N – Number of tested and evaluated products;

ε_i^l - Evaluation error, performed by expert l during evaluation of product i, $i=1,N$;

M – Number of experts, evaluating the software products;

ε_M - Error, which origins from the methodological approach, applied for the assessment of the workflow software products.

The ideal case will be when the expert identifies the quality of the software product just as its ideal assessment value A_i. Unfortunately, the background of the expert j influences the evaluation findings by his error of incompetence ε_i^j.

An addition noise influence ε_M comes from the methodology applied for the product assessment. Hence the real evaluation values concerning the quality of the product i will be:

$$RA_i = A_i + \varepsilon^l{}_i + \varepsilon_M \qquad (5.1)$$

The evaluation methodology has to minimize the influence of ε_i^l and ε_M by means the working estimations RA_i to tend towards the real value of the product quality A_i. Because the noise ε_i^l and ε_M are not measurable, during the evaluation process they have to be kept minimal if this is possible.

The approach, implemented for Task 2.2 of WP2 is based on the requirements:

- to increase the expertise of the experts, involved in the product assessment;
- to improve the evaluation methodology for assessment of the quality of the products;
- the assessment methodology to support the expert qualification in workflow product assessment

The evaluation methodology, following Chapter 4 has been based on a common standard, concerning the quality of software products. Thus, the standardization approach targets the minimization of the expert subjective influence to the evaluation findings.

The evaluation methodology has been designed to provide a common evaluation background for all experts. Thus, the noise ε_M , which arises from the methodological evaluation scheme, will be equal to all evaluation findings according to (5.1).

The next improvement in Task 2.2 evaluation scheme has been performed using the formal description (5.1). The idea in the assessment process is not to use the absolute values of the real assessment RA_i, but to make a relative comparison between the evaluated products. It means that if product *i* is assessed according to (5.1), hence the product j will have analogical assessment RA_j :

$$RA_j = A_j + \varepsilon_j^l + \varepsilon_M \ .$$

However, for making quality assessments of the products, the difference $\Delta_{i,j}$ has to be considered.

$$\Delta_{i,j} = RA_i - RA_j$$

The benefit of using $\Delta_{i,j}$ instead of RA_i and RA_j comes from the difference:

$$\Delta_{i,j} = RA_i - RA_j = A_i + \varepsilon_i^l + \varepsilon_M - (A_j + \varepsilon_j^l + \varepsilon_M) = A_i - A_j + (\varepsilon_i^l - \varepsilon_j^l).$$

Thus having the differences between the products *i, j, k* it follows:

$$\Delta_{i,j} = A_i - A_j + (\varepsilon_i^l - \varepsilon_j^l).$$

$$\Delta_{k,j} = A_k - A_j + (\varepsilon_k^l - \varepsilon_j^l). \qquad (5.2)$$

If $\Delta_{i,j} > \Delta_{k,j}$ we can strongly confirm that the software product *i* is more qualified than product *k*. This conclusion is influenced by the errors of the expert evaluations ε_i^l, ε_j^l, ε_k^l. However, having in mind that during the VISP project developments the partners experience rises, it is true that the expert errors vanish by the increase of the time:

$$\lim_{t\to\infty}\varepsilon_i^l(t) \Rightarrow 0, \qquad \forall i = 1,N, \qquad \forall l = 1,N. \tag{5.3}$$

An advantage of the classification scheme (5.2) based on relative assessments $\Delta_{i,j}$ comes from the fact that the error ε_M , originated by the evaluation scheme disappears. This is important having in mind that absolute evaluation scheme is difficult to design.

Additional benefit of scheme (5.2) in comparison with the assessment scheme (5.1) comes for the expert evaluation background. For the scheme (5.1), the final evaluation RA_i is directly influenced by the expert incompetence ε_i^l .

For the scheme (5.2), the final evaluation $\Delta_{i,k}$ is influenced by the subtraction of the two incompetences ε_i^l *and* ε_k^l . It means that if the incompetence of the evaluator *l* is the same for the different products, the integral error $\varepsilon_i^l - \varepsilon_k^l$ vanish, which is beneficial for the evaluation process.

For the case when two experts (l and m) have to assess different products, the evaluation scheme $\Delta_{i,j}$ is influenced by the incompetence of the two experts ε_i^l *and* ε_k^m . However, this incompetence is subtracted for the overall evaluation to $\Delta_{i,j}$. Thus, for the final evaluation rating $\Delta_{i,j}$ according to scheme (5.2), the incompetences of the evaluators influence slower the result when the errors are from the same sign, in comparison to the absolute evaluation scheme, residing on (1). Consequently, the relative assessment of the products, following (5.2), has three general benefits:

- the error ε_M from the evaluation methodology is suppressed;
- the evaluation findings are influenced by the difference of the evaluator's incompetence, not from their absolute values;
- for the VISP experts, according to their real work during the test of the software products, their absolute incompetence vanish:

$$\lim_{t\to\infty}\varepsilon_i^l(t) \Rightarrow 0, \qquad \forall i = 1,N, \qquad \forall l = 1,M .$$

Following these theoretical findings for the assessment of the software products, the work in D2.2 has been conducted in the following order:

- Design of a common evaluation template for the assessment of the quality of the software products;
- The evaluation template has to support the experts by decreasing their incompetence. This is achieved by adding more questions for hinting the experts in the workflow software product implementation ;
- Derive appropriate qualitative scheme and to identify scores for the quality of the products. Particularly, a quantification scheme has been developed to estimate the relative assessments RA_i , $i=1,N$ for each product. The results of these estimations have been presented as pie- chart diagrams;

- To make a unique integral score for the product quality, a weighted evaluation scheme, discussed in p.3.3.2 is applied. The evaluation findings are multiplied by the weights: 2 for week, 4 for good, 6 for strong and 0 for Not appropriate/can't assess. Thus the integral score per quality category is calculated as:

$$IRA_j = \sum_q [RA_j^q + \varepsilon_j^l + \varepsilon_M],$$

  Where RA_j^q are the evaluation findings of the expert for quality sub-category q of the product j, IRA_j is the integral score of the product j.

- Having the absolute scores IRA_i, $i=1,N$ the product quality is assessed according to a relative evaluation scheme

$$\Delta_{i,j} = IRA_i - IRA_j, \; i,j=1,N. \qquad (5.4)$$

  The products are compared each others, which gives relative assessment of the quality of the products. To minimize the relative comparisons, following relation (5.4) the product j was chosen with the lowest absolute score **min [** IRA_j **]**, j=1, N. Thus the relative assessment for the product i , $\Delta_{i,j}$, $i,j=1,N$ towards the worst product j is directly related with the absolute values of the assessment IRA_i. Hence the result of the relative assessments $\Delta_{i,j}$ can be presented as a bar chart diagram towards IRA_i and the best product will have the bigger domination over the minimal value of **min [** IRA_j], j=1,N. These considerations are used to present the relative assessment $\Delta_{i,j}$, $i,j=1,N$ of the products in a bar chart diagrams, using the absolute values of the integral product evaluation IRA_i, $i=1,N$.

- Development of new relative evaluation scheme $\Delta_{i,T}$, $i=1, N$, where T is the average result for all tested workflow software products. Thus, T represents the average level, which is supported from an ideal (but not real existing) product. The comparison of the quality of the product i towards the average value T shows the relative place of the product i concerning the current state of the technology addressing the workflow software. The results of these comparisons have been presented by a set of radar (spider) diagrams.

The common evaluation template for assessing the software product has been designed according to the chosen methodology in Chapter 4, based on a ISO/IEC 9126 standard for the quality of the software product. The template consists of the following evaluation categories:

1 General categories (G): Workflow software overview with sub-categories

  G1.1. Workflow software presentation
  G1.2. Workflow software description
  G1.3. Category of the software product
  G1.4. Supported interfaces
  G1.5. Supported standards; Confirming standards and exchange formats.

2. Functional categories (F): Principle functions

  F1. Modelling process definition
  F2. Simulation, debug

F3. Execution workflow engine
F4 Workflow client application
F5. Integration with others workflows' engines; Supported standards
F6. Administration and monitoring
FA Auxiliary functions: statistics, registration, country area information, help functionalities.

3. Reliability
4. Usability
5. Efficiency
6. Maintainability
7. Portability
8. External metrics

The different categories were decomposed when this was available to several subcategories. Thus, the evaluation scheme tries to give hints to the evaluators by means to decrease the values of these incompetences ε_i^l .

The quantitative evaluations have been introduced by the definition of four levels of assessment categories: weak, good, strong, and not applicable/cannot assess. These categories have been used to calculate scores of the products per evaluation category. The evaluation findings are given in Chapter 6.

5.3. VISP Template For Evaluation Workflow Software Products

Here is presented the evaluation template used by the partners for evaluation of the workflow software products.

Workflow software evaluation findings

I. GENERAL

DESCRIPTIONS:

Cat no	Function category	Fct no	Function name	Function implementation
G1	**Workflow software overview**	G1.1	Workflow software presentation	Workflow software title and free text presentation
		G1.2	Workflow software description	Workflow software structure and functionality – free text
		G1.3	Category of the software product	Describe the main functionalities, free text
		G.1.4	Supported interfaces	Describe the interfaces: Interface 1: Exchange of workflow

Cat no	Function category	Fct no	Function name	Function implementation
				descriptions (models, choreographies, orchestration) between the tool and other modelling tools, repositories, workflow execution tools. Interface 2: Supports the invocation of workflow engines by workflow clients. Implementation of the interface (by WSDL interface or a programming language interface) Interface 3: Application invocation, which perform additional workflow activities. Implementation of the interface (by WSDL interface, programming language interface; notification mechanism for human actions) Interface 4. Support of workflow integration and collaboration (between different domains and/or distributed workflow execution platforms Interface 5. Monitoring and administration of the workflows activities; Life cycle management of distributed workflows Additional interfaces: import/export of data towards other products like databases, file systems, repositories (UDDI, LDAP); monitoring of transactions.
		G1.5	Supported standards Conformity to standards and exchange formats	Describe: **c.** Process definition tools; Support and deviation from workflow standards; Functionality to model complex workflows; Supported user interfaces: lightweight browsers or heavy user clients. **d.** Workflow execution tools: support and deviation from workflow standards; needed technical environment; Required software product (application servers,

Cat no	Function category	Fct no	Function name	Function implementation
				supported communication protocols, transaction mechanism bindings.

II. **Functionality.** Implementation of Functions

Principal Functions

Cat no	Function category	Fct no	Function name	Function implementation
F1	**Modelling, process definition**	F1.1	Process chart: definition, process thread	Explanations how the process chart are configured as a network of processes – free text, if applicable
		F1.2	Process: description, major, minor	Presentation of the functionality of the processes, objects and their attributes – free text, if applicable
		F1.3	Event: initial, result, triggering process thread, event result correlation	Presentation of the functionality of event categories – free text, if applicable
		F1.4	Flowline	Presentation and functionalities of flow lines– free text, if applicable
		F1.5	Definition of organizational units	Presentation and functionality of organizational units and how they relate to the processes' chart– free text, if applicable
		F1.6	Junctions	Model of flow controls, if applicable
F2	**Simulation, debug**	F2.1	Objects: definition, simulation arrival profile, access data, simulation process	Description of the categories objects in the simulation framework, if applicable
		F2.2	Process: initialization, service time profile, necessary resources , available resources, child diagram	Description of the category process and its parameters, if applicable
		F2.3	Event: description, simulation event (generation, definition of object, simulation arrival profile)	Simulation functionality of object "event" , if applicable
		F2.4	Junction: description(and, or, x or), symbol, simulation outflow	Simulation functionality of object "junction" , if applicable
		F2.5	Links: simulation probability, type	Simulation functionality of object "links", if applicable

Cat no	Function category	Fct no	Function name	Function implementation
		F2.6	General conditions: simulation validation, run simulation	Support of model validation and run-time simulation, if applicable
F3	**Execution, workflow engine**	F3.1	Processes: Deployment, initialization, repositories	Peculiarities of the process deployment" , if applicable
		F3.2	Management of distributed workflows	Distributed workflow support, if applicable
		F3.3	Application Server Compatibility	Integration with Web/Application Servers, if applicable
		F3.4	File formats	Description of the suite of software files, needed for the workflow execution
		F3.5	Transaction mechanisms	
		F3.6	Supported communication protocol	Free text description
		F3.7	Users' roles	How to define different categories of users, if applicable
F4	**Workflow client applications**	F4.1	User interface technologies: client application - type, proprietary(heavy weight user clients) or general purpose(light weight browsers)	How the workflow engine operates with client applications: needs of special client software
F5	**Integration with other workflow engines; Supported standards**	F5.1	Possibilities for collaboration with other workflow engines; standards	Possibility to operate with different workflow standards
F6	**Administration and monitoring**	F6.1	Life cycle management	Possibility to manage the time schedule for modelling, simulation and execution functionalities
		F6.2	Supervising capabilities. Process status.	What kind of services, monitoring: active processes, alarm queue, Failures
		F6.3	Flexibility to workflow participants	Access rights
F7	**Other**			

Functional Auxiliary Functions

Cat no	Function category	Fct no	Function name	Function Implementation
FA1	**STATISTICS**	FA1.1	Number of users	Description, if applicable

Cat no	Function category	Fct no	Function name	Function Implementation
		FA1.2	Number of processes	Description, if applicable
		FA1.3	User rights	Description, if applicable
		FA1.4	Process Information	Description, if applicable
FA2	**REGISTRATION**	FA2.1	Country and zone registration	Description, if applicable
FA3	**Country area information**	FA3.1	User details	Description, if applicable
FA4	**HELP Functionalities**	FA4.1	Help information	Description, if applicable

Integral evaluations for the functional categories:

1. **Suitability:** the capability of the software to provide an appropriate set of functions for specified tasks and user objectives

Weak: O Good: O Strong: O Not applicable/cannot assess: O
Comments:

2. **Accuracy**: the capability of the software to provide the right or agreed results or effects

Weak: O Good: O Strong: O Not applicable/cannot assess: O
Comments:

3. **Interoperability**: the capability of the software to interact with one or more specified systems

Weak: O Good: O Strong: O Not applicable/cannot assess: O
Comments:

4. **Security:** the capability of the software to protect information and data so that unauthorized persons or systems cannot read or modify them and authorized persons or systems are not denied access to them.

Weak: O Good: O Strong: O Not applicable/cannot assess: O
Comments:

5. **Compliance:** the capability of the software to adhere to application related standards, conventions or regulations in laws and similar prescriptions.

Weak: O Good: O Strong: O Not applicable/can't assess: O
Comments:

III Reliability

This category analysis the degree, to which the tool can be trusted and what it does in unknown situation

DESCRIPTIONS:

- Robustness
  - How well does the tool behave given error conditions? Comments
- Recovery mechanisms
  - Are there checkpoints or partial recovery of entered information? Comments
  - To what degree can one recover entered data? Comments
- Consistency

- Does the tool consistently provide the same output given the same input? Comments

Integral evaluations for the reliability categories:

1. **Maturity:** the capability of the software to avoid failure as a result of faults in the software

Weak: O Good: O Strong: O Not applicable/cannot assess: O
Comments:

2. **Fault tolerance**: the capability of the software to maintain a specified level of performance in case of software faults or in infringement of its specified interface.

Weak: O Good: O Strong: O Not applicable/cannot assess: O
Comments:

3. **Recoverability:** the capability of the software to re-establish its level of performance and recover the data directly affected in the case of a failure. Following a failure, a software product will some times be down for a certain period of time, the length of which is accessed by it recoverability.

Weak: O Good: O Strong: O Not applicable/cannot assess: O
Comments:

4. **Compliance**: the capability of the software to adhere standards, conventions or regulations, relating to reliability

Weak: O Good: O Strong: O Not applicable/cannot assess: O
Comments:

IV. USABILITY

For this category of evaluation a variety of usability' characteristics are considered. Particular importance is features, daily used for this tool.

DESCRIPTIONS:

- **User Interface**
  - Does the tool provide a user interface or is it simply a conversion tool?

    YES NO: Not applicable/cannot assess:

Comments:
  - Does the user interface abide by the hardware/software system conventions?

    YES NO: Not applicable/cannot assess:

Comments:
- **User prompting**
  - Does the tool allow the user to select metadata items from lists or menus where appropriate?

    YES NO: Not applicable/cannot assess:

Comments:
  - Can an experienced user bypass those lists?

    YES NO: Not applicable/cannot assess:

Comments:
- **Minimizing duplication of entry**
  - If a provider, describing a collection can the repeatable information that remains the same across the data sets be reused and not re-entered?

    YES NO: Not applicable/cannot assess:

Comments:

- **Data Generation Integration**
    - Is the metadatabase directly accessed by data manipulation applications?

    YES NO: Not applicable/cannot assess:

Comments:
    - Does the tools support workflow planning - status recording, auditing and notification. (Eg e-mail or report to data owner that a data set requires updating according to its metadata).

    YES NO: Not applicable/cannot assess:

Comments:
- **Cut and Paste**
    - Can standard metadata information items be pulled or pasted in from external sources?

    YES NO: Not applicable/cannot assess:

Comments:
    - Can metadata elements be saved for use in other metadata documents?

    YES NO: Not applicable/cannot assess:

Comments:
- **Restart-ability**
    - Does the tool allow metadata to be written in stages?

    YES NO: Not applicable/cannot assess:

Comments:
    - Does it support incremental updates?

    YES NO: NOT APPLICABLE:
- **Documentation**
    - Is documentation provided?

    YES NO: Not applicable/cannot assess:

Comments:
    - Does the tool provide on-line help?

    YES NO: Not applicable/cannot assess:

Comments:
    - Does the tool's help function use the help mechanism native to the hardware/software system (and presumably familiar to the user?)

    YES NO: Not applicable/cannot assess:

Comments:
    - How extensive is the documentation?

    Comments:
    - How extensive is the on-line help?

    Comments:
- **Miscellaneous**
    - Does the tool provide a spell checking capability?

    YES NO: Not applicable/cannot assess:

Comments:
    - Can new terms be added to the spell checking dictionary?

    YES NO: Not applicable/cannot assess:

Comments:
    - Does the tool provide a mechanism for maintaining a history of metadata changes?

    YES NO: Not applicable/cannot assess:

Comments:

Integral evaluations for the usability categories:

1. Reuse of Meta data elements from similar documented data sets.

Weak: O Good: O Strong: O Not applicable/cannot assess: O
Comments:

2. Possibility to compare the tool to other applications in a same environment

Weak: O Good: O Strong: O Not applicable/cannot assess: O
Comments:

3. Availability and completeness of documentation and on-line help

Weak: O Good: O Strong: O Not applicable/cannot assess: O
Comments:

4. Understandability: the capability of the software product to enable the user to understand whether the software is suitable and how it can be used for particular tasks and conditions of use; this will depend on the documentation and initial impressions given by the software.

Weak: O Good: O Strong: O Not applicable/cannot assess: O
Comments:

5. Learnability: the capability of the software product to enable the user to learn its application

Weak: O Good: O Strong: O Not applicable/cannot assess: O
Comments:

6. Operability – the capability of the software product to enable the user to operate and control it

Weak: O Good: O Strong: O Not applicable/cannot assess: O
Comments:

7. Attractiveness: the capability of the software product to be liked by the user

Weak: O Good: O Strong: O Not applicable/cannot assess: O
Comments:

8. Compliance: the capability of the software to adhere to standards, conventions, style guides or regulations, relating to usability

Weak: O Good: O Strong: O Not applicable/cannot assess: O
Comments:

V. EFFICIENCY

Integral evaluations for the efficiency categories:

1. **Time behaviour**: the capability of the software to provide appropriate response and processing times and throughput rates when performing its function, under stated conditions

Weak: X Good: O Strong: O Not applicable/cannot assess: O
Comments:

2. **Resource utilization**: the capability of the software to use appropriate resources in appropriate time when the software performs its function under stated conditions

Weak: X Good: O Strong: O Not applicable/cannot assess: O
Comments:

3. **Compliance**: the capability of the software to adhere to standards or conventions relating to efficiency

Weak: X Good: O Strong: O Not applicable/cannot assess: O
Comments:

4. **Metadata exchange** The tool is evaluated for its support for metadata import, generated by other tools or export of metadata in a standard form that would be ingestible by other tools.

- **Import capabilities**
  - Does the tool can import existing metadata for revision?

    YES NO: Not applicable/cannot assess:
    Comments:
  - Are the input formats it will accept flexible?

    YES NO: Not applicable/cannot assess:
    Comments:
  - What are those formats?

    Comments:
  - Does the tool can import metadata elements from external sources, e.g., a DBMS?

    YES NO: Not applicable/cannot assess:
    Comments:
  - What hooks are provided to connect the tool with external sources of information?

    Comments:

- **Export capabilities**

  What format does the tool generate?
  Comments:
  - Is there support for multiple output formats?

    YES NO: Not applicable/cannot assess:
    Comments:

- **Completeness / Compliance**
  - Is the output complete / compliant with workflow standard formats?

    YES NO: Not applicable/cannot assess:
    Comments:

VI. MAINTAINABILITY

This category of evaluation considers acquisition, installation, maintenance issues that would normally concern a system administrator or someone installing the tool.

Integral evaluations for the maintainability categories:

1. **Analyzability:** the capability of the software product to be diagnosed for deficiencies or causes of failures in the software, or for the parts to be modified to be identified

Weak: X Good: O Strong: O Not applicable/cannot assess: O
Comments:

2. **Changeability:** the capability of the software product to enable a specified modification to be implemented; Implementation includes coding, designing and documenting changes.

Weak: X Good: O Strong: O Not applicable/cannot assess: O
Comments:

3. **Stability:** the capability of the software to minimize unexpected effects from modifications of the software

Weak: X Good: O Strong: O Not applicable/cannot assess: O
Comments:

4. **Testability:** the capability of the software product to enable modified software to be validated

Weak: X Good: O Strong: O Not applicable/cannot assess: O
Comments:

5. **Compliance**: the capability of the software to adhere to standards or conventions relating maintainability

Weak: X Good: O Strong: O Not applicable/cannot assess: O
Comments:

6. **Administrative**: This category of evaluation criteria considered acquisition, installation, and maintenance issues that would normally concern a system administrator or someone installing the tool. We also tried to consider how adaptable a tool would be as the metadata standard changed or whether a user would have to rely on the support of the tool's creators.

- **Platforms / Installation**
  - What hardware/software platforms does the tool operate on?

Comments:

  - Does the tool work on more than one hardware/software combination?

Comments:

  - How much disk space and memory does the tool require?

Comments:

  - How easy is the tool to install?

Comments:

- **Stand-alone**
  - Is the tool dependent on other software packages or libraries?

YES: NO: Not applicable/can't assess:
Comments:

  - Are these provided?

Comments:

  - Does the tool cause any conflicts with other common software packages on the platform

YES: NO: Not applicable/can't assess:

Comments:

- Is the tool publicly available?

YES: NO: Not applicable/can't assess:

Comments:

- Is it maintained?

YES: NO: Not applicable/can't assess:

Comments:

- What is the cost of acquiring the tool?

Comments:

- **Updates**
  - Is there a mechanism to modify / update the tool or the input or output it generates?

YES: NO: Not applicable/can't assess:

Comments:

- Can an administrator, for example, add new menu items or change the order of presentation?

YES: NO: Not applicable/can't assess:

Comments:

- How will the tool be updated when the standard is updated?

Comments:

- Will administrators be able to modify the tool appropriately?

YES: NO: Not applicable/can't assess:

Comments:

- How easy is that?

Comments:

- Is source code provided?

YES: NO: Not applicable/can't assess:

Comments:

VII. PORTABILITY

This category considers acquisition, installation and maintenance issues that would normally concern a system administrator or someone, installing the tool.

Integral evaluations for the portability categories:

1. **Adaptability:** the capability of the software to be modified for different specified environments without applying actions or means other than those provided for this purpose for the software considered.

Weak: O Good: O Strong: O Not applicable/can't assess: O

Comments:

2. **Installability:** the capability of the software to be installed in a specified environment

Weak: O Good: O Strong: O Not applicable/can't assess: O

Comments:

3. **Co-existence:** the capability of the software to co-exist with other independent software in a common environment, sharing common resources

Weak: O Good: O Strong: O Not applicable/can't assess: O
Comments:

4. **Replaceability**: the capability of the software to be used in place of other specified software in the environment of the software. Replaceability is used in place of compatibility in order to avoid possible ambiguity with interoperability. Replaceability does not imply that this software is able to replace the software under consideration.

Weak: O Good: O Strong: O Not applicable/can't assess: O
Comments:

5. **Compliance:** the capability of the software to adhere to standards or conventions, relating to portability

Weak: O Good: O Strong: O Not applicable/can't assess: O
Comments:

VIII. EXTERNAL METRICS

Software Evaluation Checklist:

Elements **Information**

General information	
Product name	Comments
Producer	Comments
Price options	Comments
Open platform	Comments
Platform specific	Comments
Server Purchased	Comments
Hosted	Comments
License fee	Comments
Content covered	Comments
Intended audience(s)	Comments
Documentation (provided/useful)	Comments
Program free of bugs & errors	Comments

Entry level skills required (describe)	Comments
Computer skills required (describe)	Comments
Other technical skill requirements (describe)	Comments
Distribution policy: installation	Comments
Distribution policy: download	Comments
Market share and position in the market: maintenance	Comments
Market share and position in the market: technical support	Comments

5.4. Conclusions

The overview performed for the methodologies used for the evaluation of software products and information resources demonstrates that the evaluation methodology can be strongly influenced by subjective reasons. These influences can origin from both the evaluation methodology, the choice of the criteria, the incompetence of the experts, and the users' expectations from the software product. To minimize these subjective influences, this work gives preferences to the standardization approach by performing a comparison of the product according to the recommendations of the standard ISO/IEC9126 for assessing the quality of the software product. Using the main categorization scheme for quality assessment of software product an evaluation template is developed. Thus using common evaluation scheme the evaluation works the deliverable targets to overcome the general drawback that different evaluator have to evaluate different software products. The common evaluation scheme and the derived template are prerequisites for minimizing the evaluation errors. A comparative evaluation scheme is developed. It allows minimizing the evaluation errors, originating from the methodological drawbacks of the evaluation scheme and from the personal incompetence of the different evaluator. A relative assessment and comparison is introduced. Thus, the quality of the products has been assessed by:

- Absolute evaluation, notated as $\boldsymbol{RA_i}$**.** Pie chart diagrams define these quality assessments;
- An absolute integral evaluation $\boldsymbol{IRA_i}$ is applied according to weighted assessment scheme. Thus a unique value per quality category is found for each software product;
- A relative evaluation between the products is applied, notated as Δ_{ij}. A block diagrams are defined for these relative assessments, which compare the products towards the minimal value of $\boldsymbol{IRA_i}$, $i=1,N$;
- A relative evaluation of all products towards the average level of the product quality is evaluated. The results of these comparisons are given as radar diagrams.

The evaluation findings and their graphical interpretation are presented in Chapter 6 of the deliverable D2.2.

6. Evaluation Findings

6.1 Introduction

It is virtually difficult to give a full overview and a detailed comparative analysis for all practically available software products and the reason that the best software product can be found in this prospective raising domain as workflow management and business process automation. It is, however, a useful exercise to make a bit general comparison of prospective products since the evaluation procedure may otherwise get lost in details and fail to see the main objective and essential characteristics of each product.

The presentation in this chapter is therefore not a brief summary of each product description, but rather a snapshot of each product's most noticeable aspects from our own point of view. We tried to establish a common basis for comparison and evaluation of the products and to derive charts for comparative presentation. The more details of each product are given in the general description for each, performed in chapter3 of the deliverable. Here for each product it is presented:

- A pie-chart which illustrates how well each product meets the main requirements for the software quality ;
- A bar-chart, which compares the average scores of the products in the category where they compete
- A radar diagram, which assess the integral software quality of the product

The two next sections provide

- an evaluation of the functional characteristics of the products by means to identify their application classes, supported standards and availability for business implementations;
- evaluation and appraise the products by giving scores of the software quality of the products, following a methodology for multicriteria evaluation and comparisons

6.2. Functional Evaluation Of Characteristics Of Short-List Products

Description

The functional evaluation of the short-list products in alphabetical order is presented bellow. The analysis is based on the assessments provided by the partners. The evaluations are divided into six general criteria, related to the main software quality categories: functionality, reliability, usability, efficiency, maintainability and portability. In this multicriteria evaluation scheme of this stage of the VISP developments, the consortium gave preferences on the functional peculiarities of the products. This particular weighted choice is explained with the needs of the VISP consortium to have possibility to model and to manage different web services, which are targeted by the project. Thus, the results of the functional evaluation are provided in this chapter and the companion evaluations towards the other five criteria are provided in the annex. The evaluations of the functionalities of the products have qualitative nature. They explain and summary the standardization background of the product, its features to cooperate with other software products, the possibility to model, simulate, manage and

administer the workflow management processes. The evaluation findings are result from installation, configuration and trial test of the chosen software among the set of candidate technologies. Thus, the evaluation is given by direct test with the product, which improves the partner experience both in working with this class of software products as with their ability to manage problems, related to the design of workflow management systems.

An additional outcome from the tests and personal evaluations is the integral evaluation of the products, addressing the main software quality characteristics. This integral assessing introduces the quantitative scores of the quality of the products. The integral evaluation has been performed by giving expert ranking for every software quality sub criteria. Four-level scale has been chosen: week, good, strong, cannot assess/not applicable, which formalize the expert opinion for the appropriate quality sub-criteria. A particular evaluation finding for the main quality criteria "functionality" is presented below.

Integral evaluations for the functional categories:

1. **Suitability:** the capability of the software to provide an appropriate set of functions for specified tasks and user objectives

Weak: O Good: O Strong: ✓ Not applicable/can't assess: O
Comments: BPEL 1.1 implementation with strong extensions

2. **Accuracy**: the capability of the software to provide the right or agreed results or effects

Weak: O Good: O Strong: ✓ Not applicable/can't assess: O
Comments: Correct BPEL 1.1 implementation

3. **Interoperability**: the capability of the software to interact with one or more specified systems

Weak: O Good: ✓ Strong: O Not applicable/can't assess: O
Comments: Further evaluation necessary

4. **Security:** the capability of the software to protect information and data so that unauthorized persons or systems cannot read or modify them and authorized persons or systems are not denied access to them.

Weak: O Good: O Strong: O Not applicable/can't assess: **?**
Comments:

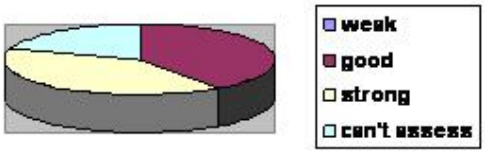
5. **Compliance:** the capability of the software to adhere to application related standards, conventions or regulations in laws and similar prescriptions.

Weak: O Good: ✓ Strong: O Not applicable/can't assess: O
Comments: Compliant to BPEL 1.1 / WSDL 1.1 but with proprietary extensions

By making a relative assessment of this software product, the vector evaluation for the category "functionality" is:
FUNCTIONALITY: (2 from 5) "strong"; (2 from 5) "good"; (1 of 5) "can't assess" or these evaluation findings can be expressed in percentage in vector form as:

FUNCTIONALITY: 0% "weak"; 40% good; 40% "strong"; 20% "Not applicable/can't assess".

This result is illustrated as a pie- chart diagram, which for the case is presented as:



Similar pie - chart diagrams are defined for every integral software quality criteria: functionality, reliability, usability, efficiency, maintainability and portability. Thus a relative evaluation of the products, in regards to it different software characteristics is found.

To obtain a common evaluation score for each product, a weighted evaluation scheme is applied. To each evaluation level of the software sub criteria, a weighted coefficient is assigned. This weight corresponds to the relatively worth of the evaluation level. For this study the weights, which were applied were chosen to be:

- week: score 2;
- good: score 4;
- strong: score 6;
- Cannot assess/not-applicable 0

This weighted evaluation scheme allows the vector evaluation of the main software quality characteristics to be allocated a unique quantitative level. Thus, the "functionality" evaluation criteria will have the score:
FUNCTIONALITY: 2 x 0% + 4 x 40% + 6 x 40% + 0 x 20% = 4.0

6.3 Evaluations

ActiveBPEL Engine

I. Functionality

Principal Functions

F1 Modelling, process definition

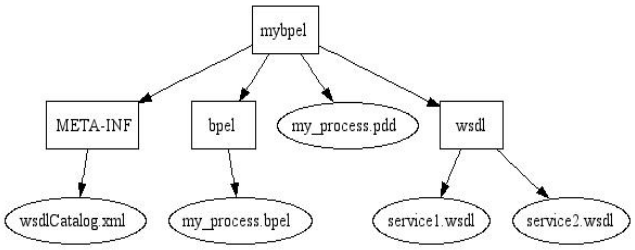
F1.1	Process chart: definition, process thread	Processes are directly deployed in the engine using BPEL files.
F1.2	Process: description, major, minor	Deployed processes are graphically displayed by Administration tool using a graph representation and are described with BPEL and WSDL standards.
F1.3	Event: initial, result, triggering process thread, event result correlation	All processes transactions, results and alarms are visualized into Admin console.
F1.4	Flowline	Not applicable.
F1.5	Definition of organizational units	Standard BPEL 2.0 processes description.
F1.6	Junctions	Standard BPEL 2.0 processes description.

F2 Simulation, debug

F2.1	Objects: definition, simulation arrival profile, access data, simulation process	The engine supports workflow simulation and debugging but only in the ActiveWebflow professional version, which will be bought apart, and it runs only in Windows based systems. By this application, the user can view all real time interactions and engine messages. Moreover, it is possible to change flow and processes parameters during the simulation, and messages sent by the engine are displayed into a box console.
F2.2	Process: initialization, service time profile, necessary resources, available resources, child diagram	The simulation needs only to define a sample set of data for its execution. At this point, you can start simulation and execute step by step the designed workflow. The simulation will be represented by a workflow graph panel, and the current activity will be clearly underlined.
F2.3	Event: description, simulation event (generation, definition of object, simulation arrival profile)	Event can be simulated directly using activities, which had been defined into BPEL file. Those can be modified by user at simulation time.
F2.4	Junction: description(and, or, x or), symbol, simulation outflow	Junctions can be simulated and changed in real time.
F2.5	Links: simulation probability, type	During the simulation, a user can modify or correct all links. Data, which is possible to change, are name, source, and target and transition condition.
F2.6	General conditions: simulation validation, run simulation	Not applicable

F3 Execution, workflow engine

F3.1	Processes: Deployment, initialization, repositories	Deploying a BPEL process involves creating a deployment archive file (a JAR with an extension of ".bpr") and copying that to your servlet container. To create this archive, you need to organize your files into a particular directory structure, create one or two configuration files, and then create an archive from that directory. Create a directory for your deployment files; we'll name it mybpel in this example. Create the subdirectories • bpel • META-INF • wsdl • partners (optional) As an example, let's say you have one BPEL file my_process.bpel and two WSDL files service1.wsdl and service2.wsdl. Your directory structure would look something

		like this:  Using this directory structure, create the archive and copy it to your servlet container. Your WSDL files could live anywhere, even on another machine. Packaging them inside the .pdd lets the ActiveBPEL engine get to them quicker. The ActiveBPEL engine reads the WSDL references from the .pdd file and uses the location attribute of the <wsdl> element as a key into the WSDL catalog. If the WSDL catalog contains a matching location, the engine loads the WSDL from the corresponding classpath. If no mapping exists in the catalog, the engine assumes the location is an absolute URL and attempts to load the WSDL from that location. The engine supports hot deployment, thus administrators can add new processes or modify existing processes without restarting it.
F3.2	Management of distributed workflows	Not applicable
F3.3	Application Server Compatibility	The engine is able to work on all Web/Application Servers, which are compatible with J2EE specifications, although our tests were performed on Apache Tomcat only.
F3.4	File formats	For the workflow execution three files are necessary: • BPEL • WSDL • WSDL catalog which provides a way for the ActiveBPEL engine to find WSDL Optionally, a user can give the PDD file for process deployment description.
F3.5	Transaction mechanisms	Transactions are supported by Web Service - Transaction standards. The Web Services Transactions specifications define mechanisms for transactional interoperability between Web services domains and provide a means to compose transactional qualities of service into Web services applications.
F3.6	Supported communication protocol	SOAP and ASOAP communication protocols are fully supported.
F3.7	Users' roles	User roles are defined using BPEL standard. Thus, a user can define roles using keyword "partnerRole" in BPEL source files.

F4 Workflow client applications

F4.1	User interface technologies: client application - type, proprietary (heavy weight user clients) or general purpose (light weight browsers)	The workflow engine operates with client applications using web services communication or java API invocations.

F5 Integration with other workflow engines; Supported standards

F5.1	Possibilities for collaboration with other workflow engines; standards	The engine is only compatible with BPEL 1.x or 2.0 standards.

F6 Administration and monitoring

F6.1	Life cycle management	All parameters for life cycle management are editable in ActiveWebFlow application, in fact using the engine administration tool is not possible to manage life cycle.
F6.2	Supervising capabilities Process status	Using the administration tool a user can supervise processes status and alarm queue. The tool shows a graph, which represents the process and updates it in real time.
F6.3	Flexibility to workflow participants	The BpelAdmin is accessible to all workflow partecipants; authentication management tools are not provided with ActiveBPEL engine distribution.

Functional Auxiliary Functions

FA1 Statistics

Fct no	Function name	Function Implementation
FA1.1	Number of users	Not possible
FA1.2	Number of processes	Active processes are visible in administration tool and a user can view all processes, which are in execution or stopped onto engine. It is possible to define selection filter criteria to display certain execution period or certain process state (running, completed, faulted).
FA1.3	User rights	Not possible
FA1.4	Process Information	All processes information is showed in the process status panel.

FA2 HELP Functionalities

Help is accessible directly by Administration tool. It is a simply set of html pages and explain well Administration tool possibilities and how to configure the engine and persistent storage.

II Integral Characteristics

	Active BPEL Engine				
Characteristics ISO/IEC 9126	**weak**	**good**	**strong**	**can't assess**	**score result**
Functionality	20%	20%	60%		**4,8**
Reliability			75%	25%	**4,5**
Usability		50%	12,5%	37,5%	**2,75**
Efficiency		67%		33%	**2,68**
Maintainability		60%	20%	20%	**3,6**
Portability		20%	60%	20%	**4,4**

III Pie- Charts

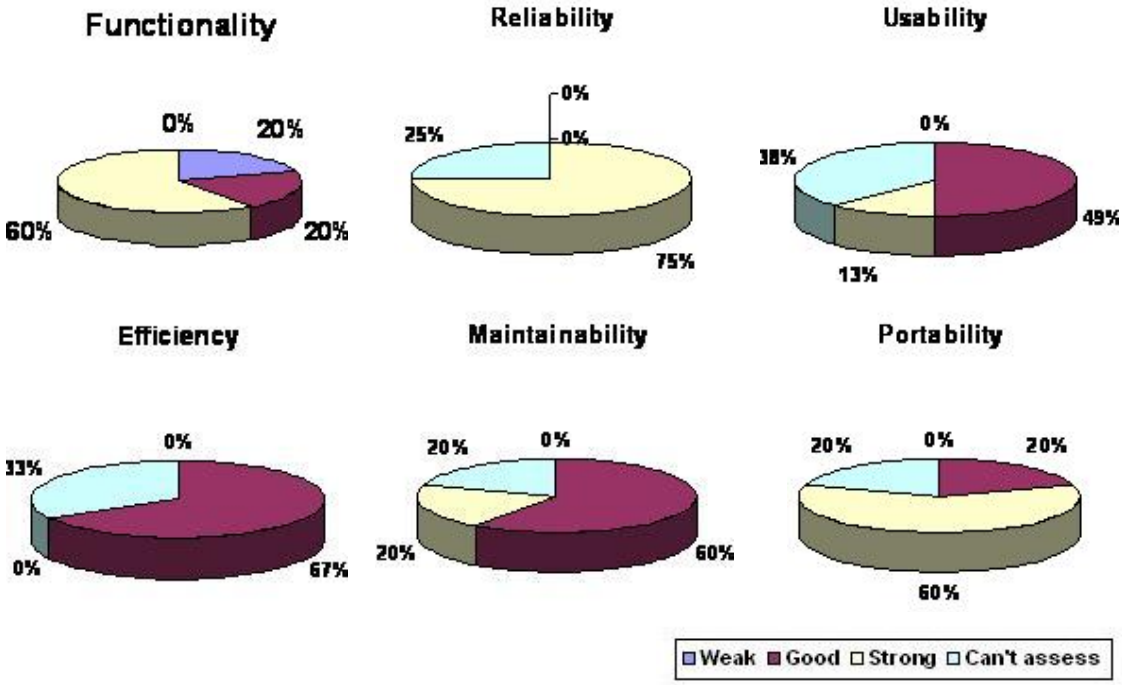


Figure 6.1 Pie-charts for Active BPEL Engine's characteristics

Active Endpoints ActiveWebflow Standard

I. Functionality

Principal Functions

F1 Modelling, process definition

Fct no	Function name	Function implementation
F1.1	Process chart: definition,	BPEL compliant

Fct no	Function name	Function implementation
	process thread	
F1.2	Process: description, major, minor	See above
F1.3	Event: initial, result, triggering process thread, event result correlation	-
F1.4	Flowline	-
F1.5	Definition of organizational units	-
F1.6	Junctions	-

F2 Simulation, debug

Not applicable

F3 Execution, workflow engine

F3.1	Processes: Deployment, initialization, repositories	A business process archive is a container for all the relevant files needed for deploying a BPEL process to the ActiveWebflow server. It is an archive file with a .bpr extension and is similar to a Web archive file. " .bpr" files can be created in two ways: • Automatically create a .bpr file in ActiveWebflow Designer. • Manually create a .bpr file. The basic business process archive includes: BPEL process (.bpel file), WSDL file(s) referenced in the BPEL process (.wsdl file), WSDL catalog (.xml file), Process deployment descriptor (.pdd file), and optionally, a partner definition file (.pdef file).
F3.2	Management of distributed workflows	Based on BPEL, WSDL, SOAP
F3.3	Application Server Compatibility	ActiveWebflow Standard installs into the Apache Tomcat Server
F3.4	File formats	The following files BPEL process (.bpel file), WSDL file(s) referenced in the BPEL process (.wsdl file), WSDL catalog (.xml file), Process deployment descriptor (.pdd file), and optionally, a partner definition file (.pdef file), are packed into a .bpr file and then the process is ready to be deployed.
F3.5	Transaction mechanisms	WS Transactions
F3.6	Supported communication protocol	SOAP
F3.7	Users' roles	BPEL defines several constructs to identify roles and relationships used in interactions.

F4 Workflow client applications

F4.1	User interface technologies: client application - type, proprietary(heavy weight user clients) or general purpose(light weight browsers)	Generic SOAP client or JAVA API

F5 Integration with other workflow engines; Supported standards

F5.1	Possibilities for collaboration with other workflow engines; standards	**AWS** supports BPEL standard for choreography and orchestration of services and WSDL/SOAP conventions, so the collaboration should be based on these standards.

F6 Administration and monitoring

F6.1	Life cycle management	Web-based consoles allow the configuration and administration of the server's execution environment and deployed and running processes. The Engine console pages allow the main settings of the engine to be viewed and managed: • Enable and disable process instance logging • Turn on and off message validation • Enable or disable BPEL extensions • Set engine caching parameters • Add and remove licenses • Prune completed processes by date • Prune deployment logs by date • Prune deployed process definitions • Version and build numbers for all ActiveWebflow Standard components The Deployment console pages allow the deployed processes and their artifacts to be managed: • Deployment of Business Process Archive (.bpr) files • Deployment logs for each .bpr file for errors and warnings • Lists of all deployed processes with selection filters • Detailed version information including the ability to modify specific data based on the version state (e.g., status, effective and expiration dates) • Detailed information regarding other artifacts deployed with process • WSDL files deployed in the BPR • Global Partner definition files deployed to the server
F6.2	Supervising capabilities. Process status.	The Process Status console pages allow to view and manage information associated with individual process instances. These pages provide the ability to sort and filter the information in ways that make easier to deal with only

		the most relevant process information. For running and completed processes the consoles are used in order to perform the following tasks: • Suspend, Resume and Terminate executing processes • View the process execution state graphically, hierarchically, and textually • Examine the content and status of process variables, activities and links For the persistent queues that allow for the re-hydration of processes on server startup the following can be inspected: • Alarms -- events that will execute at a specific time as specified by executing processes • Receives -- the result of a BPEL activity or Event Handler specifying that a message is expected Once a process is deployed to the AWS server, the administrator can leverage the capabilities of ActiveWebflow Professional to debug and analyze the execution of running or completed processes.
F6.3	Flexibility to workflow participants	Access rights

Functional Auxiliary Functions

FA1 Statistics

Fct no	Function name	Function Implementation
FA1.1	Number of users	Not applicable
FA1.2	Number of processes	The statistic is available in the Administration Console
FA1.3	User rights	Not applicable
FA1.4	Process Information	Full information about the deployed processes is available in the Administration Console

FA2 HELP Functionalities

There are two User Guides with very useful and thorough help information.

II Integral Characteristics

	ActiveWebflow™ Standard				
Characteristics ISO/IEC 9126	**weak**	**good**	**strong**	**can't assess**	**score result**
Functionality		20%	60%	20%	**4,4**
Reliability		50%	25%	25%	**3,5**
Usability		12,5%	75%	12,5%	**5**
Efficiency		67%		33%	**2,68**
Maintainability		20%		80%	**0,8**
Portability		20%	20%	60%	**2**

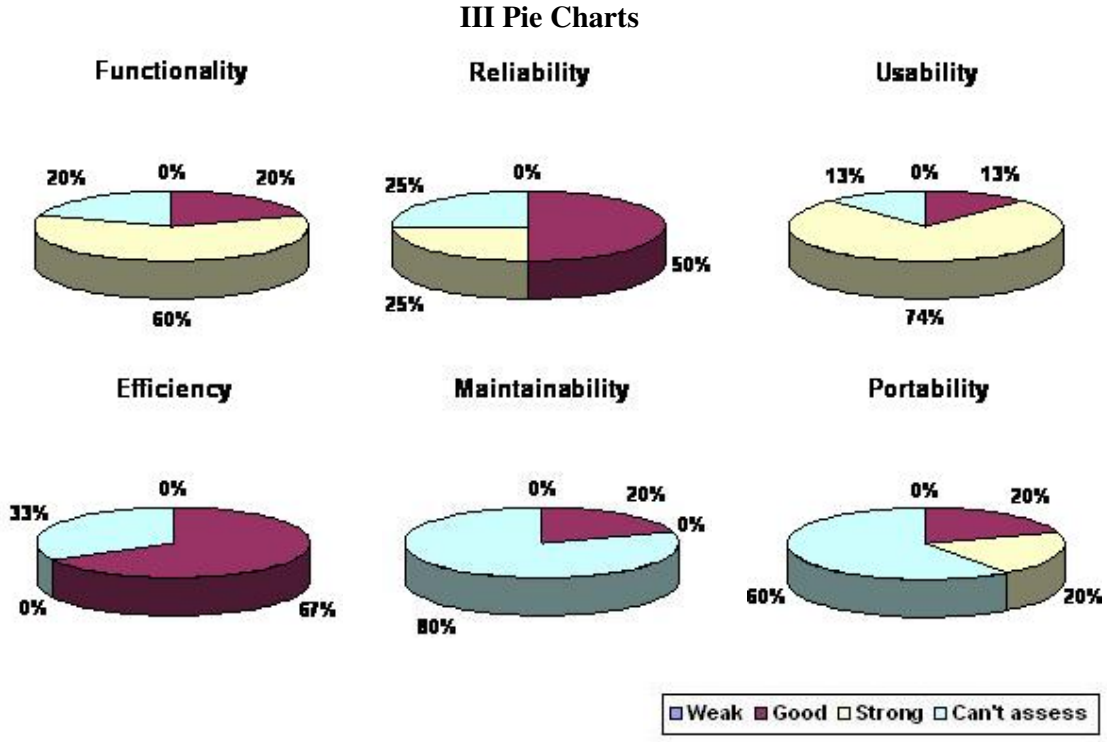


Figure 6.2 Pie-charts for ActiveWebflow Standard's characteristics

ActiveWebflow Designer

I. Functionality

Principal Functions

F1 Modelling, process definition

Fct no	Function name	Function implementation
F1.1	Process chart: definition, process thread	A BPEL process has several structural parts: • Process Element and Properties • Variables • Partner Links • Activities • Fault Handlers • Compensation Handlers **Process Properties** - Process Name, Query Language, Target Namespace **Variables** - The BPEL process receives, manipulates, and sends data through XML variables. Variables are defined in one of the following ways: • WSDL message types • XML schema simple types • XML schema elements **Partner Links –** used when defining Web service

Fct no	Function name	Function implementation
		interaction activities. A partner link definition consists of: • A unique name • A partner link type • At least one role and can include two roles • Role names of *My Role*, which is the process role, and *Partner Role* At a minimum, a process must have one partner link for each type of conversation it is engaged in. **Activities -** the processing steps, performed in the order described by the flow diagram. In ActiveWebflow, there are several types of activities: • Basic activities to exchange and assign data and define execution steps • Containers to structure activities and handle faults and compensation • Event Handlers **Fault Handlers.** A fault handler defines the activity that the process must perform in response to an error condition. **Compensation Handlers** Compensation is the process of reversing or providing an alternative for a successfully completed activity, especially when a fault occurs.
F1.2	Process: description, major, minor	Two kinds of processes can be built: abstract and executable. ActiveWebflow provides containers to structure activities. Each container has its own rules for how its child activities are executed.
F1.3	Event: initial, result, triggering process thread, event result correlation	A BPEL process instance has a beginning, middle, and end, which are all defined by activities. The process must begin with one of the following: • Receive activity • Pick activity • Flow activity containing one or more Receive or Pick activities The middle of the process consists of linked activities, running in sequence or concurrently. The process instance ends in one of the following ways: • An activity defines that the process is complete • A fault reaches a process scope, and the process terminates abnormally
F1.4	Flowline	**Link** A link is a structure that connects two activities and controls the order of execution of the two activities. A link can include a condition, further controlling when, or if, an activity executes. The condition is an XPath expression.

Fct no	Function name	Function implementation
F1.5	Definition of organizational units	There are basic activities and structured activities.
F1.6	Junctions	Model of flow controls, if applicable In Active Webflow Designer, these are Links with a Transition Condition. One can control the start of a link-targeted activity by writing an expression for a join condition property.

F2 Simulation, debug

F2.1	Objects: definition, simulation arrival profile, access data, simulation process	Processes can be simulated within ActiveWebflow, using sample data, or remotely debugged from the ActiveBPEL server. During simulation, one can debug the process by using breakpoints or stepping. In simulate/run mode, the process executes but the execution cannot be suspended or examined. In simulate/step mode, execution can be suspended and resumed and variables can be inspected and modified.
F2.2	Process: initialization, service time profile, necessary resources , available resources, child diagram	When one simulates execution of a process in ActiveWebflow, the simulation engine behaves identically to the server engine, where the deployed processes run. It can be set preferences to modify the server engine behaviour, and the same preferences for the simulation engine. **Prerequisites for Simulation** • all the required properties for each activity are filled in • the Abstract Process property is set to No for the process • Make sample data available for all variables • If the process contains a Pick activity, you can select a branch to execute
F2.3	Event: description, simulation event (generation, definition of object, simulation arrival profile)	As the process executes, the following events occur: Each activity is highlighted as it prepares to execute • In Process Variables view, sample data is cleared at the beginning of simulation and then is displayed as the corresponding activities execute • The Console view shows simulation events • If an error occurs, an error message pops up, the simulation terminates, and the activity is marked with an X. When a simulation terminates, the process diagram shows the simulation path. One can step through the execution of a BPEL process one activity at a time. While you are stepping through the simulation, execution results can be seen in the Console. **Viewing the Execution State of an Activity or Link**

		During simulation and remote debugging, BPEL activities and links can have one of the following states: • Inactive • Active (links only) • Ready to execute • Executing • Dead Path (will not run because of links or conditional execution) • Execution Completed (activity only) • Execution Faulted (activity only)
F2.4	Junction: description (and, or, x or), symbol, simulation outflow	See above
F2.5	Links: simulation probability, type	See above
F2.6	General conditions: simulation validation, run simulation	Not applicable

F3 Execution, workflow engine
Not applicable

F4 Workflow client applications
Not applicable

F5 Integration with other workflow engines; Supported standards
Not applicable

F6 Administration and monitoring
Not applicable

Functional Auxiliary Functions

FA1 Statistics
Not applicable

FA2 Help Functionalities
AWDesigner User Guide is very useful and thorough help provided as a part of Active Webflow Designer.

II Integral Characteristics

	ActiveWebflow™ Professional Designer				
Characteristics ISO/IEC 9126	**weak**	**good**	**strong**	**can't assess**	**score result**
Functionality			60%	40%	**3,6**
Reliability			75%	25%	**4,5**
Usability		25%	62,5%	12,5%	**4,75**
Efficiency		67%		33%	**2,68**
Maintainability	20%	20%		60%	**1,2**
Portability		40%		60%	**1,6**

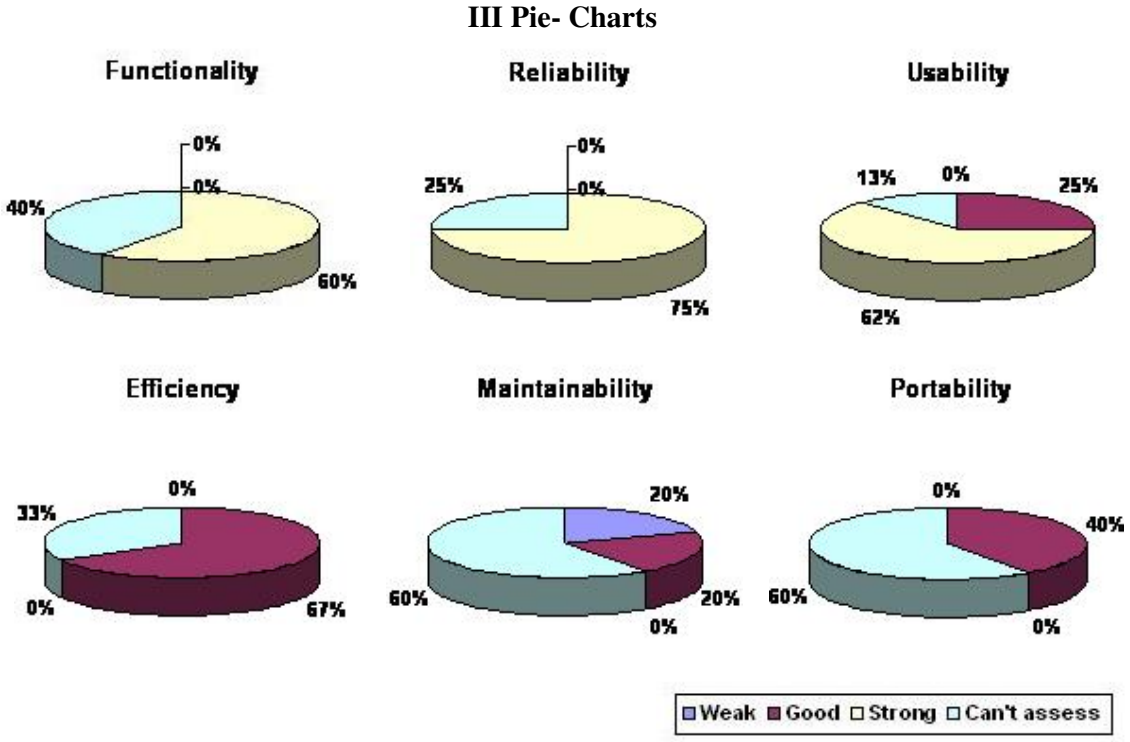


Figure 6.3 Pie-charts for ActiveWebflow Designer's characteristics

BizTalk Server

I. Functionality

Principal Functions

F1 Modelling, process definition

Fct no	Function name	Function implementation
F1.1	Process chart: definition, process thread	Process charts can be defined in two different ways : either with Visio (and the appropriate set of diagram tools) or with Visual Studio 2005. The Visio additional toolbox is designed for Business Modellers and it offers a high view of the process with basic flow (Activity, Junction, parallel activity, Loop). Visual Studio designer is designed for developers. This is where the process designed in Visio will be implemented (definitions of Ports, Send/Receive activities, Exceptions). The process definition can be exchanged between those two programs, without any loss of information: both provide Import/Export functionality for that purpose.
F1.2	Process: description, major, minor	Processes are defined as a flow of activities that are "drag and drop'ed" from the toolbox to the designer panel. Several activities are provided to interact with the workflow such as Send/Receive. A workflow can also start another workflow.

Fct no	Function name	Function implementation
F1.3	Event: initial, result, triggering process thread, event result correlation	A "Listen" activity enables to arm a timer and execute a flow of activities if no message is received.
F1.4	Flowline	Flow lines don't hang properties.
F1.5	Definition of organizational units	Presentation and functionality of organizational units and how they relate to the processes' chart– free text, if applicable
F1.6	Junctions	A "Decide" activity is provided and behaves as a "If then Else" structure. It is not possible to define multiple conditions so another "Decide" activity must be inserted in the "Else" branch to add another rule.

F2 Simulation, debug

F2.1	Objects: definition, simulation arrival profile, access data, simulation process	The Health and Activity Tracking (HAT) tool provides features for reporting, analyzing, and debugging data and messages that are archived in the BizTalk Tracking database.
F2.2	Process: initialization, service time profile, necessary resources , available resources, child diagram	
F2.3	Event: description, simulation event (generation, definition of object, simulation arrival profile)	
F2.4	Junction: description (and, or, x or), symbol, simulation outflow	
F2.5	Links: simulation probability, type	
F2.6	General conditions: simulation validation, run simulation	

F3 Execution, workflow engine

F3.1	Processes: Deployment, initialization, repositories	Process deployment is done with the contextual menu in Visual studio. Deployment is automated but if the project you are trying to deploy uses ports that are already in use, you must first disable them.
F3.2	Management of distributed workflows	Distributed workflow support, if applicable
F3.3	Application Server Compatibility	N/A
F3.4	File formats	Definition schemas are saved in .xsd files. Orchestration definitions are stored in .odx files (XML file).
F3.5	Transaction mechanisms	Scopes can be defined, as well as Exceptions and Compensation Activities. Transactions are either Atomic or Long Running.
F3.6	Supported communication protocol	Communication Protocols supported are: - POP3/SMTP - SOAP, HTTP - FILE - MSMQ

		- MQ Series Following B2B languages are also supported : - RosettaNet - HIPAA - HL7 - SWIFT
F3.7	Users' roles	"Role Link" feature in the designer enables the grouping of Ports.

F4 Workflow client applications

F4.1	User interface technologies: client application - type, proprietary(heavy weight user clients) or general purpose (light weight browsers)	Clients can be of various kind: Net client, any client aware of Web Service (if the business process is deployed as such), mail clients, etc.

F5 Integration with other workflow engines; Supported standards

F5.1	Possibilities for collaboration with other workflow engines; standards	A process deployed on BizTalk can be part of a collaboration workflow (support for BPEL, XLANG).

F6 Administration and monitoring

F6.1	Life cycle management	
F6.2	Supervising capabilities Process status	Monitoring is done with BAM (Business Activity Monitoring) portal solution. You can then search for a process instance and display specific information about it, setup alerts.
F6.3	Flexibility to workflow participants	N/A

Functional Auxiliary Functions

FA1 Statistics
Not applicable

FA2 HELP Functionalities
Help information

II Integral Characteristics

	BizTalk Server				
Characteristics ISO/IEC 9126	**weak**	**good**	**strong**	**can't assess**	**score result**
Functionality	20%	60%		20%	**2,8**
Reliability	25%	50%		25%	**2,5**
Usability	12,5%	37,5%	25%	25%	**3,25**
Efficiency	33%	34%		33%	**2,02**
Maintainability		60%	20%	20%	**3,6**
Portability	80%			20%	**1,6**

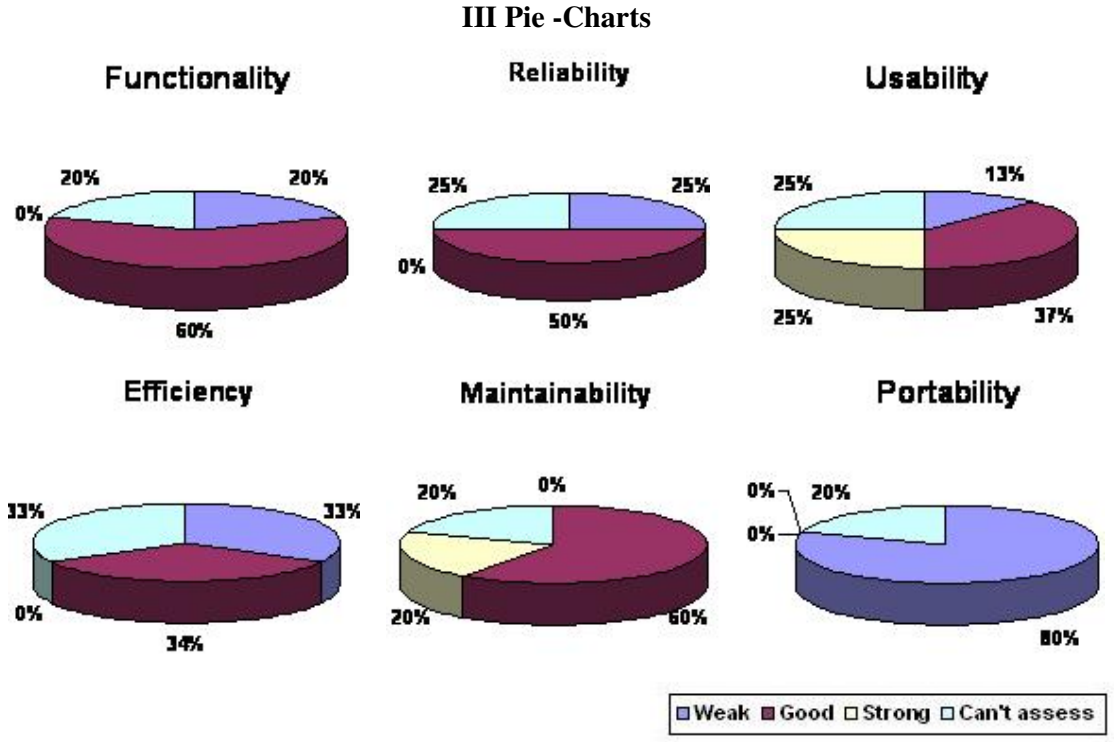


Figure 6.4 Pie-charts for Biztalk Server's characteristics

Borland Together Designer 2006

I. Functionality

Principal Functions

F1 Modelling, process definition

Fct no	Function name	Function implementation
F1.1	Process chart: definition, process thread	Borland Together Designer 2006 contains a user-friendly graphical modelling editor. The business process diagram tools palette in the editor provides all necessary elements for modelling business activities. The user can drag and drop elements to a BPMN diagram and set after that their properties in a special window. Together supports BPMN, with some minor changes related to mapping BPMN to BPEL. Together provides support for the most frequently needed diagrams and notations defined by the UML 1.4. It supports the following UML diagrams: • Component diagram; • Use case diagram; • Deployment diagram; • Activity diagram; • Sequence diagram; • Collaboration diagram;
F1.2	Process: description, major, minor	
F1.3	Event: initial, result, triggering process thread, event result correlation	
F1.4	Flowline	
F1.5	Definition of organizational units	
F1.6	Junctions	

Fct no	Function name	Function implementation
		• Statechart diagram; • Class diagram (includes packages). Together also provides support for the most frequently notations defined by the UML 2.0 and supports the following diagrams: • Activity; • Class; • Component; • Composite structure; • Deployment; • State Machine; • Use case; • Interaction (Sequence and Communication) Diagrams.

F2 Simulation, debug

F2.1	Objects: definition, simulation arrival profile, access data, simulation process	Borland Together Designer 2006 provides a special view for the BMPN validation. After the validation, the tool creates a list of problems with information about potential reasons. From the context menu, the user can quickly find on the diagram the element, which could not pass the validation.
F2.2	Process: initialization, service time profile, necessary resources , available resources, child diagram	
F2.3	Event: description, simulation event (generation, definition of object, simulation arrival profile)	
F2.4	Junction: description (and, or, x or), symbol, simulation outflow	
F2.5	Links: simulation probability, type	
F2.6	General conditions: simulation validation, run simulation	

F3 Execution, workflow engine
Not applicable

F4 Workflow client applications
Not applicable

F5 Integration with other workflow engines; Supported standards
Not applicable

F6 Administration and monitoring
Not applicable

Functional Auxiliary Functions

FA1 Statistics

Fct no	Function name	Function Implementation
FA1.1	Number of users	One user

Fct no	Function name	Function Implementation
FA1.2	Number of processes	not applicable
FA1.3	User rights	not applicable
FA1.4	Process Information	not applicable

FA2 HELP Functionalities

Borland Together Designer 2006 provides comprehensive documentation and examples for the business process modelling on BPMN.

II Integral Characteristics

	Borland Together Designer 2006 for Eclipse				
Characteristics ISO/IEC 9126	**weak**	**good**	**strong**	**can't assess**	**score result**
Functionality	20%	60%	20%		**4**
Reliability	50%	50%			**3**
Usability		62,5%	37,5%		**4,75**
Efficiency		33%		67%	**1,32**
Maintainability	40%	20%		40%	**1,6**
Portability	20%	40%	40%		**4,4**

III Pie- Charts

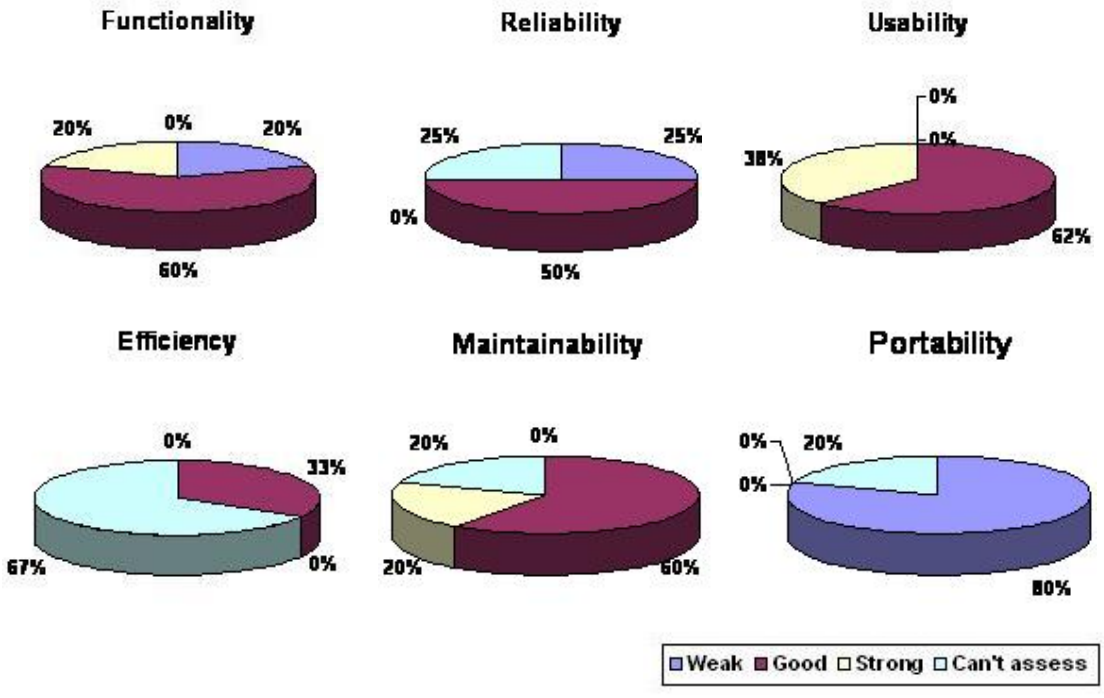


Figure 6.5 Pie-charts for Borland Together Designer's characteristics

Cape Clear

I Functionality

Principal Functions

F1 Modelling, process definition

Fct no	Function name	Function implementation
F1.1	Process chart: definition, process thread	CapeClear 6.5 provides a very handy graphical BPEL process designer. Both text and graphical versions can be displayed. Three different views are available in the designers: the activity view (where the main process is defined), the faults view (where fault handlers are defined), and the Event view. Many activities are available for defining a business process. Those are grouped in four categories : - Communication : activities available are Receive, Invoke, Reply, Pick (defines possible sequences for a list of events; when one of an event is trigger, the corresponding diagram is processed), OnMessage and OnAlarm (both events, first one is similar to Receive activity, the second marks a time out event which is trigger after a given time) - Control : Switch, Case, Otherwise (all three to implement a "If-Then-ElseIf-Else" structure), While and Wait - Fault : Throw (throws an error), Catch (catch faults) and CatchAll (catch faults not previously caught), Terminate (terminates the process instance) - Other: Assign, Copy (Assign is a combination of Copy activities to handle local variables values), Sequence, Empty, Scope (defines a sub process)
F1.2	Process: description, major, minor	Processes can be of two types: • Invoke, receive and reply messages: this can be achieved in two ways; either drags and drops an external operation from the partner link container into the designer panel, or drags and drops the appropriate activity in the diagram and chooses the correct parameters. • Handling data: this is done with the assign Activity. It's possible to assign a local variable from an expression, literal data, incoming messages… Inside the same assign activity can be processes multiple copies.
F1.3	Event: initial, result, triggering process thread, event result correlation	Events are defined in the Event view of the process designer. Events can be of two types : • OnMessage: the diagram process defined in the event handler is processed when the defined message is triggered. This can be used when a user wants to cancel the current task. • OnAlarm: an alarm is armed, either "until" or "for" an amount of time (can be defined in months, days, hours)
F1.4	Flowline	Flow lines cannot be parameterized. They only define a possible path for the business process.
F1.5	Definition of organizational units	The scope activity is a good way to define sub process. You can then define events handlers and Fault handlers for the sub

Fct no	Function name	Function implementation
		process. However, it is not possible to reuse a defined scope.
F1.6	Junctions	Flow controls are the following : - Switch case and otherwise: this is used to perform an if-else-if control. First, the switch activity must be dragged in the process window. It defines by default a Case activity and an Otherwise activity. One can specify as many case activities as one wants. The evaluation is made for every Case statement until one return 'true': then the sequence of activities defined in case statement is processed. If no evaluation statement returns 'true', then the sequence of activities in the Otherwise node is processed.

F2 Simulation, debug

F2.1	Objects: definition, simulation arrival profile, access data, simulation process	Cape Clear 6.5 does not provide Debugging support. However, it is possible to change the default log level to log SOAP messages (requests and replies).
F2.2	Process: initialization, service time profile, necessary resources , available resources, child diagram	N/A
F2.3	Event: description, simulation event (generation, definition of object, simulation arrival profile)	N/A
F2.4	Junction: description(and, or, x or), symbol, simulation outflow	N/A
F2.5	Links: simulation probability, type	N/A
F2.6	General conditions: simulation validation, run simulation	The BPEL file is validated every time it's saved in the process designer. It's validated another time while being deployed on the server.

F3 Execution, workflow engine

F3.1	Processes: Deployment, initialization, repositories	When you create a new Web service project, Cape Clear Studio creates a ws-application.xml file in the /ws/WSAR-INF directory of the project. The project's ws-application.xml file is the application descriptor for the server. This file contains static information about the application WSDL, schema, mappings, and component or adapter type. Default values for these elements are configured automatically by Cape Clear Studio when generating the ws-application.xml file. However, one can change several of these settings through the WS Application Editor user interface. Some of the ws-application.xml file configuration

		details cannot be changed after deployment, so one must make changes to the configuration before deploying the Web service.
F3.2	Management of distributed workflows	Cape clear 6.5 workflows supports either being called in a more global workflow or call other distant workflow.
F3.3	Application Server Compatibility	CapeClear 6.5 engine can be run on the following application servers : • jBoss • WebLogic • Websphere • Tomcat
F3.4	File formats	The WSDL files describing the business process as well as the partners services, are stored under the /ws directory of the project. The BPEL script is stored in the /WS/WSAR_INF directory of the project with a ".bpel" extension. In the same subdirectory is the ws_application.xml file in which can be configured the URL of the WSDL file this service implements, the URL of the BPEL file, whether to load or not the service if the validation of the BPEL file does not succeed.
F3.5	Transaction mechanisms	Scopes can be defined so that exceptions can be caught in a compensation block, but remote transactions aren't supported. Note that compensation is not supported in the runtime of the tested version (6.5).
F3.6	Supported communication protocol	Cape Clear 6.5 supports a various number of communication protocols: • SNMP: for administration • JMS (Java Messaging Service) for asynchronous messaging (bundled server) • HTTP, FTP, SMTP (mail), file drop • SOAP (for web services) over HTTP/FTP. • Websphere MQ • WS-RM (WS-Reliable Messaging)
F3.7	Users' roles	N/A

F4 Workflow client applications

F4.1	User interface technologies: client application - type, proprietary(heavy weight user clients) or general purpose(light weight browsers)	Cape Clear 6.5 runs the BEPL process as a Web service. It is then possible to invoke the process from any client supporting web services (web clients, java stand-alone clients). Note that Cape Clear 6.5 provides a very useful tool inside the Cape Clear Studio: it can dynamically generate SOAP requests to test deployed business process. The screen is split in half: the first half contains the request that the developer can update with its own data (random data is generated by default). The second half is the SOAP answer from the service.

F5 Integration with other workflow engines; Supported standards

F5.1	Possibilities for collaboration with other workflow engines; standards	Cape Clear 6.5 supports workflow collaboration: as the process is seen as a web service, it can be re used by any of other workflow systems supporting web services. Moreover, it is very easy to design an orchestration with the graphical process designers starting from the WSDL of collaborated workflows. Each partner can be identified by a given colour in the diagram, which helps a lot for readability.

F6 Administration and monitoring

F6.1	Life cycle management	The Cape Clear Manager, a web based administration application, enables the administrator to deploy, edit (like disable a web service), or even suppress a deployed Web Service.
F6.2	Supervising capabilities Process status	The Cape Clear Orchestration Manager lists all the deployed BPEL processes, giving detailed information such as status (green/red light); deployed time, active, suspended, completed, terminated and failed process instances. Clicking on the count of the chosen count gives the details.
F6.3	Flexibility to workflow participants	Both Cape Clear Manager and Cape Clear Orchestration Manager access are protected with a login/password.

Functional Auxiliary Functions

FA1 Statistics

Fct no	Function name	Function Implementation
FA1.1	Number of users	-
FA1.2	Number of processes	Processes are listed.
FA1.3	User rights	-
FA1.4	Process Information	Each process can be clicked to get detailed information (process instances)

FA2 HELP Functionalities
Help information

II Integral Characteristics

	Cape Clear 6.5				
Characteristics ISO/IEC 9126	**weak**	**good**	**strong**	**can't assess**	**score result**
Functionality		40%	40%	20%	**4**
Reliability	25%	50%		25%	**2,5**
Usability	12,5	50%	25%	12,5%	**3,75**
Efficiency		67%		33%	**2,68**
Maintainability	20%	40%		40%	**2**
Portability	20%	40%	20%	20%	**3,2**

III Pie Charts

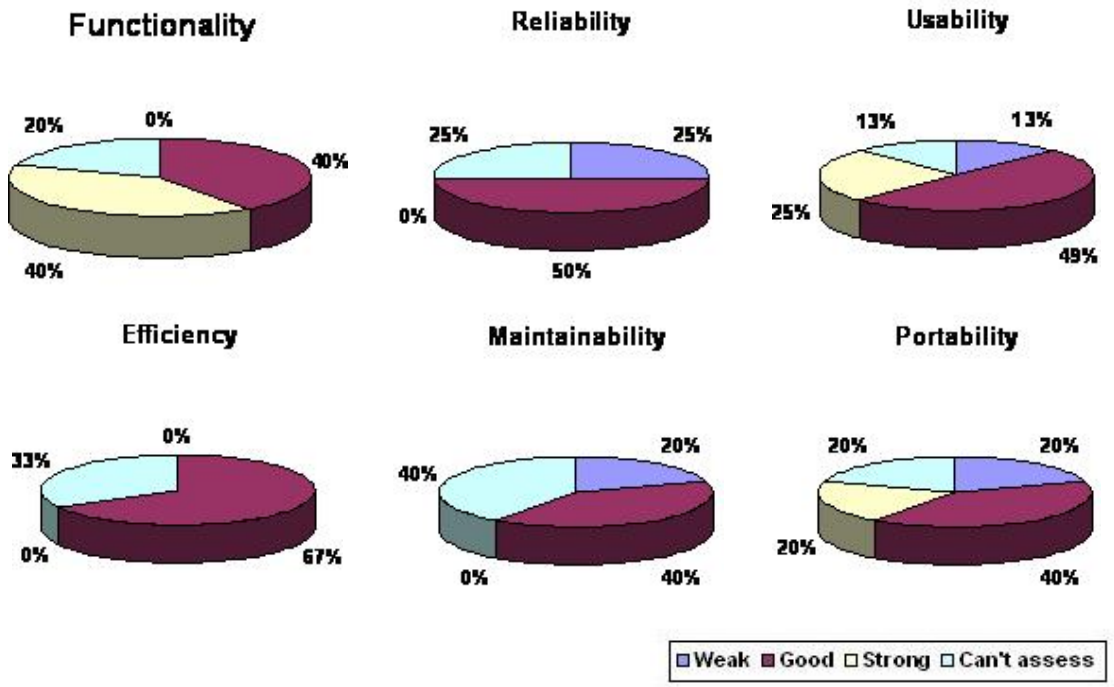


Figure 6.6 Pie-charts for Cape Clear's characteristics

Con-cern

I. Functionality

Principal Functions

F1 Modelling, process definition

Fct no	Function name	Function implementation
F1.1	Process chart: definition, process thread	A process is a loosely coupled collection of activities, which are executed on a subject (e.g. proposal, application, reclamation,) depending on its state. A project can contain one or more processes with or without interconnections. An activity is an isolated task (transaction), executed on a subject. The activity obtains information from the subject and / or adds information to the subject and interacts with the environment. If the execution of an activity executes incompletely or with errors, it will be re-executed after a delay. A distinction is drawn between synchronous and asynchronous activities. A synchronous activity is invoked by the controller, as soon as its preconditions are met. The result is available immediately. If the preconditions of an

Fct no	Function name	Function implementation
		asynchronous activity are met, it is enlisted for asynchronous execution. An actor (user / other process / event) may fetch the subject from the list, manipulate it and announce the change at any time.
F1.2	Process: description, major, minor	A process is described as a set of activities with pre- and post-conditions. An activity is executed when its preconditions are met. It manipulates the process item, thereby creating postconditions.
F1.3	Event: initial, result, triggering process thread, event result correlation	An event is basically an asynchronous activity without a precondition. It is executed by an external trigger, for example a scheduler timeout. A listener is basically a synchronous activity without a postcondition. It is called by the controller, when its preconditions are met. Listeners can destroy the subject.
F1.4	Flowline	The process flow is determined at run-time. The graphical modelling tool generates the edges (arrows) between the activities or events and the conditions, according the description of pre and post-conditions in the activity properties.
F1.5	Definition of organizational units	The process chart is composed by activities, conditions, events, listeners, edges, and operators (and/or).
F1.6	Junctions	The and/or operators are related to the pre/post-conditions and other junction types are not allowed

F2 **Simulation, debug**

F2.1	Objects: definition, simulation arrival profile, access data, simulation process	Not applicable
F2.2	Process: initialization, service time profile, necessary resources , available resources, child diagram	Not applicable, except the Gantt chart generated using the graphical modelling tool
F2.3	Event: description, simulation event (generation, definition of object, simulation arrival profile)	It is described by a Java class. A template is generated from the process description.
F2.4	Junction: description(and, or, x or), symbol, simulation outflow	It is described by a Java class. A template is generated from the process description.
F2.5	Links: simulation probability, type	Simulation functionality of object “links”, if applicable
F2.6	General conditions: simulation validation, run simulation	Support of model validation and run-time simulation , if applicable

F3 Execution, workflow engine

F3.1	Processes: Deployment, initialization, repositories	Depends on the external application server
F3.2	Management of distributed workflows	Not a subject for con:cern
F3.3	Application Server Compatibility	Runs on any J2EE-compliant application server
F3.4	File formats	Java classes, appropriate JARs
F3.5	Transaction mechanisms	
F3.6	Supported communication protocol	SOAP, JMS, JCA
F3.7	Users' roles	How to define different categories of users, if applicable

F4 Workflow client applications

F4.1	User interface technologies: client application - type, proprietary or general purpose(light weight browsers)	The clients are written in Java using the con:cern API. The web site presents three examples of clients.

F5 Integration with other workflow engines; Supported standards

F5.1	Possibilities for collaboration with other workflow engines; standards	There are no clear statements concerning the possibilities to collaborate with other workflow engines.

F6 **Administration and monitoring**

F6.1	Life cycle management	There are no facilities to manage the time schedule for modelling, simulation and execution functionalities
F6.2	Supervising capabilities. Process status.	There are no supervising capabilities.
F6.3	Flexibility to workflow participants	Access rights

Functional Auxiliary Functions

FA1 Statistics

Not applicable

FA2 HELP Functionalities

Not applicable

II Integral Characteristics

	con:CERN				
Characteristics ISO/IEC 9126	**weak**	**good**	**strong**	**can't assess**	**score result**
Functionality	60%	20%		20%	**2**
Reliability	50%	50%			**3**
Usability	50%	25%	12,5%	12,5%	**2,75**
Efficiency	67%	33%			**2,66**
Maintainability	20%	20%	20%	40%	**2,4**
Portability		20%	80%		**5,6**

III Pie Charts

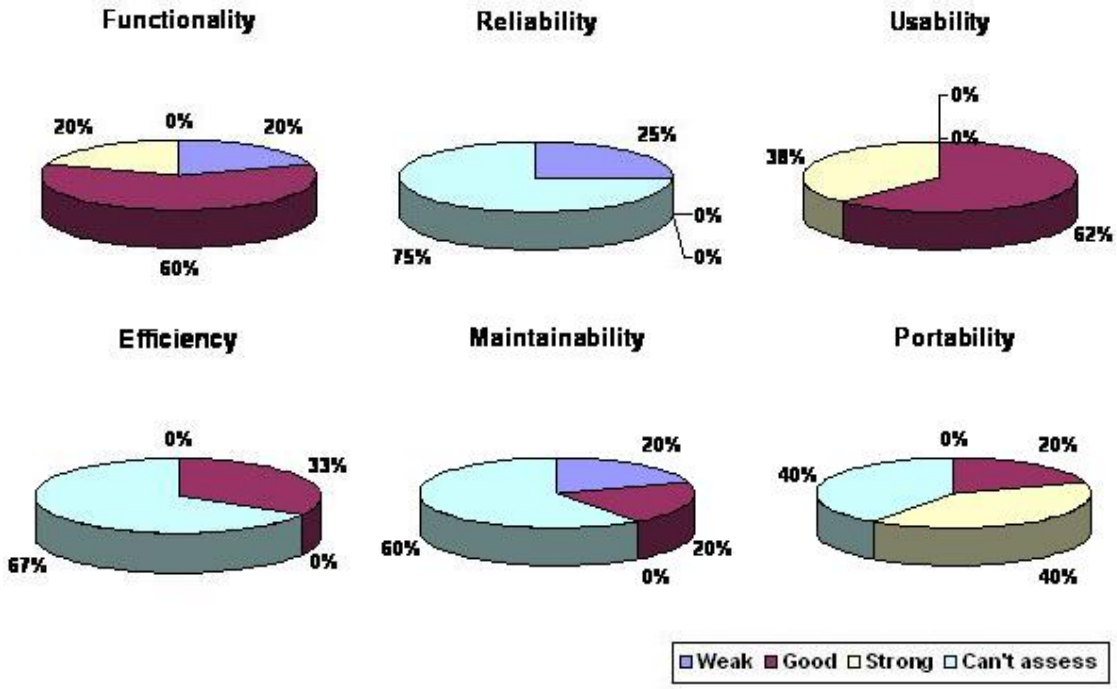


Figure 6.7 Pie-charts for Con Cern's characteristics

Enhydra JaWE

I Functionality

Principal Functions

F1 Modelling, process definition

Fct no	Function name	Function implementation
F1.1	Process chart: definition, process thread	JaWE is consisted of two logical parts - "Package level" and "Process level". Package level: • Meta-Model - describes the top-level entities

Fct no	Function name	Function implementation
		contained within a Process Definition, their relationships and attributes. • Package - It is possible to define several processes within one package, which may share the same tools (applications) and participants. The Package acts as a container for grouping together a number of individual process definitions and associated entity data, which is applicable to all the contained processes' definitions. - Package Attributes • Workflow Application Declaration - is a list of all applications or tools required and invoked by the workflow processes. - Formal Parameters - are parameters that are interchanged with the application via the invocation interface. - External Reference - is a reference to an external definition of an entity. • Workflow Relevant Data - represent the variables of a Process Definition or Package Definition. They are typically used to maintain decision data or reference data values, which are passed between activities or sub-processes. • Workflow Participant Specification (Minimal Organisation Model) - it is defined by a type and related information, which is a set of type specific attributes. - LDAP Support • Declared Data Types - WfMC assumes a number of standard data types; such data types are relevant to workflow relevant data, system, environmental data, or participant data. Expressions may be formed using such data types to support conditional evaluations. • External Package - External package reference allows referencing definitions contained within other Package definitions. - Importing processes - JaWE provides a possibility of importing processes from external packages. • Extended Attributes - Extended Attributes are the primary method to support such extensions. They are attributes those defined by the user or vendor, where necessary, to express any additional entity characteristics. Process level: • Workflow Process Definition - The Process Definition entity provides contextual information that applies to other entities within the process. It is a container for the process itself and provides information associated with the administration or it is used during process execution. • Workflow Applications Declarations – is a list of all

Fct no	Function name	Function implementation
		applications or tools required and invoked by the workflow process. • Workflow Relevant Data - represent the variables of a Process Definition or Package Definition. • Workflow Participants - All descriptions and explanations for participants at Package level are almost the same as on the Process level. • Workflow Process Activity - is a piece of work that will be done by combination of resources and computer applications. - Atomic Activities - Subflow - Block Activity - Route - Start and End Objects • Transitions - Link between two activities is established by transitions. They describe possible transitions between activities and the conditions that enable or disable them during workflow execution. JaWE has three types of transition - simple, self-routed and circular.
F1.2	Process: description, major, minor	At package level, user can choose among several XPDL views: graphical, text and xpdl view which represents the XPDL, as it will be saved into file. Graphical view of package level is divided into some parts. Left side of the main window shows Package Hierarchy Tree. Root of the tree is main package and branches are imported external packages. If these external packages have their own external packages, they are shown in the tree too. Since all packages have unique IDs, circular references are allowed, though tree expansion will stop on package, which is displayed before. Right side of the main window displays workflow processes defined in package, which is selected on the hierarchy tree. Processes are presented only symbolic with not so many information about process and with no graphical layout of process elements (activities and transitions). Double click on any process will open workflow-editing window - Process level. The actions represented by buttons are applied to the currently selected package within the package hierarchy tree. Other parts of Package level are title bar, menus, few toolbars, status bar and tabs for selecting package view. The second part of JaWE is Process level, which is some kind of editing window. This part of JaWE is used for graphical representation of process definition and for defining attributes of entities on that level.

Fct no	Function name	Function implementation
		When selected Graph view, the main portion of workflow editing window takes working area. Here user insert visible objects and adjust them. First thing drawn must be a participant. When at least one participant is made visible, user can insert other elements. These elements are activities and transitions. • Atomic activities(Generic) • Subflow activity • Block activity • Route activity Route activity does not implement any action. It is used for synchronization and conditional branching, so it shares common attributes with generic. JaWE has three types of transition - simple, self-routed and circular.
F1.3	Event: initial, result, triggering process thread, event result correlation	Workflow process consists of a number of workflow activities. • Atomic activities(Generic) Majority of activities are atomic (Generic Activity), in which case they are the smallest units of work, although even atomic activity may produce more than one work item for its performer, or may invoke more than one application. • Subflow activity Subflow is another activity type. It implements a completely new workflow process. Process definition within the subflow is entirely independent from the first one (where subflow activity resides). It has its own set of activities, internal transitions, participants, application definitions and other workflow relevant data. Latter three though may be inherited from the package, which is common for both workflow process definitions. • Block activity An activity may be a block activity that executes an activity set, or map of activities and transitions. Activities and transitions within an activity set share the name space of the containing process. • Route activity Dummy (route) activity does nothing on its own. This type of activities is used for synchronization and constructing complex and sophisticated transitional conditions - e.g. activity pre- and post- conditions.
F1.4	Flowline	See F.1.6
F1.5	Definition of organizational units	Workflow process consists of a number of workflow activities. • Atomic activities(Generic) • Subflow activity • Block activity

Fct no	Function name	Function implementation
		• Route activity See also F1.3
F1.6	Junctions	Link between two activities is established by transitions. "Transitions" describe possible transitions between activities and the conditions that enable or disable them during workflow execution. JaWE has three types of transition - simple, self-routed and circular. Simple transition is link between two activities, represented graphically with one straight line. Self-routed transition is link between two activities which is graphically 'broken' in three parts (but essentially, they do not differ in the XPDL logic they are representing), and circular transition is transition from activity to itself, and is graphically represented as a circle with an arrow.

F2 Simulation, debug
Not applicable

F3 Execution, workflow engine
Not applicable

F4 Workflow client applications
Not applicable

F5 Integration with other workflow engines; Supported standards
Not applicable

F6 Administration and monitoring
Not applicable

Functional Auxiliary Functions
FA1 Statistics
Not applicable

FA2 HELP Functionalitis
Support

II Integral Characteristics

	Enhydra JaWE				
Characteristics ISO/IEC 9126	**weak**	**good**	**strong**	**can't assess**	**score result**
Functionality		60%	20%	20%	**3,6**
Reliability			25%	75%	**1,5**
Usability		62,5%	12,5%	25%	**3,25**
Efficiency		100%			**4**
Maintainability		20%		80%	**0,8**
Portability	20%	40%		40%	**2**

III Pie Charts

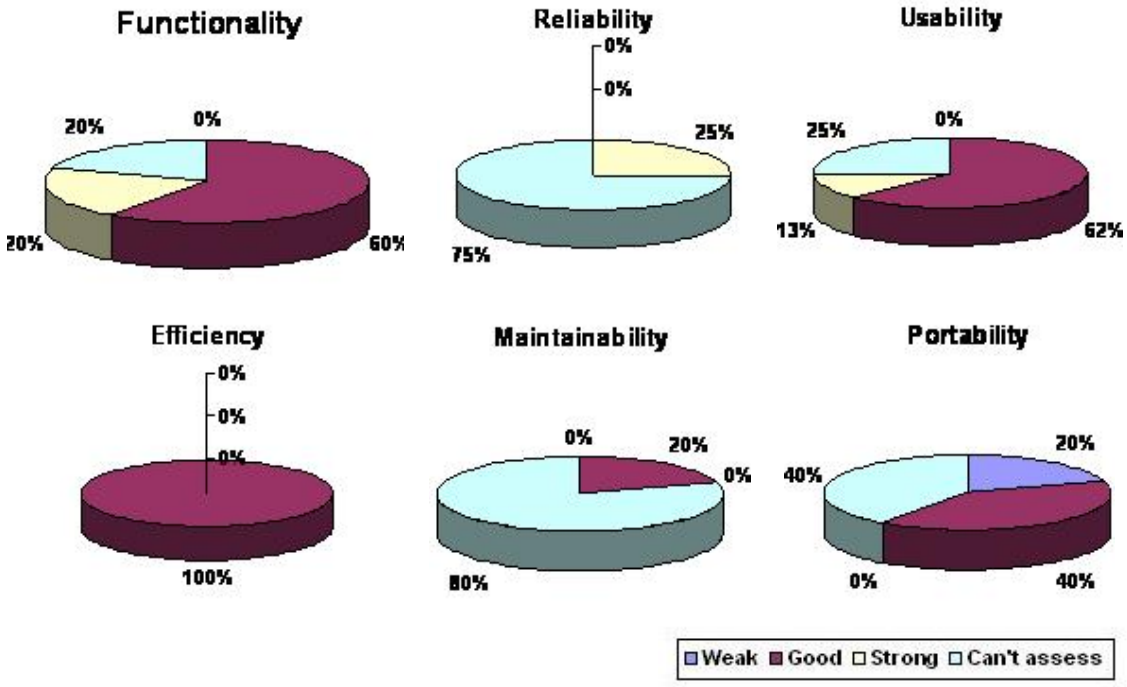


Figure 6.8 Pie-charts for Enhydra JaWE's characteristics

Enhydra Shark

I. Functionality

Principal Functions

F1 Modelling, process definition

Not applicable (Shark only features the WF engine). However, a complete modeler (Jawe) is available that fully integrates with Shark.

F2 Simulation, debug

It is not applicable to Shark itself. Shark, via its monitoring application, can provide detailed information about a process' status.

F3 Execution, workflow engine

Fct no	Function name	Function implementation
F3.1	Processes: Deployment, initialization, repositories	It supports repository management, package management, process instantiation management. The workflow XPDL representation can be loaded into shark engine using the administrator application. User can load xpdl files into the workflow system repository for process instantiation. Moreover, the Wf-XML protocol is supported.
F3.2	Management of	Unsupported. Can be implemented using Web

Fct no	Function name	Function implementation
	distributed workflows	Services, RMI.
F3.3	Application Server Compatibility	The Shark workflow engine consists essentially of a Java library. Applications built on top of it can be hosted by any Application Server (examples provided with Shark run on Apache Tomcat).
F3.4	File formats	XPDL, XML
F3.5	Transaction mechanisms	Transactions are supported by Shark transaction manager. This component implements and manages persistency of transactions state using a Enhydra DODS relational-object-mapping. Transactions are mapped into SharkTransaction objects and are visible to the client, and to the internal shark interfaces. Moreover, it is possible to use Shark with external transaction.
F3.6	Supported communication protocol	Shark supports SOAP, ASAP (Asynchronous SOAP), Corba, Wf-XML. Due to the exposing of API interfaces, other protocols can be implemented if necessary.
F3.7	Users' roles	Shark supports user roles mechanisms with its Authentication and UserGroup API. The first one is DB based (in fact, it uses DB for storing and retrieving information about organizational structure), and the other one is LDAP based (it uses LDAP server for getting organizational information). Supports user management • Accounts - manage the users of the shark server by defining the new ones, deleting the existing ones or changing their properties. • Logged - displays the list of currently logged users • Mapping - enables mapping the package and package's processes participants to the real shark users.

F4 Workflow client applications

F4.1	User interface technologies: client application - type, proprietary(heavy weight user clients) or general purpose(light weight browsers)	Along with its POJO interface, it provides a CORBA interface through which the CORBA client applications can communicate with the shark deployed as a CORBA service. It can be configured to use organizational structure defined on LDAP server (with specific implementation of shark's UserGroup and Authentication component). Its interfaces allow passing of "external" transactions (used in some applications), so shark can work with this "client" transactions. It uses DODS (OR/M tool from Enhydra), which enables shark to use almost any DB system for storing information, and it can be easily configured to switch target DB vendor and/or url (it has predefined scripts, and means to automatically create appropriate

		tables in those DBs using Octopus - ETL tool from Enhydra). Shark can use custom Java classes (and even interfaces or abstract classes) as process variables. Supported user interfaces: heavy user client, lightweight client.

F5 Integration with other workflow engines; Supported standards

F5.1	Possibilities for collaboration with other workflow engines; standards	No explicit mechanism is presented to collaborate with other workflow engines.

F6 Administration and monitoring

F6.1	Life cycle management	The Administration tool manages all main functionalities for the execution of the workflow processes, while modelling and simulation can be managed using an external tool (e.g. JaWE).
F6.2	Supervising capabilities. Process status.	Process instantiation management: Active processes, Name, Category, Version, State (Enable/Disable). Process monitor: Start, Suspend, Resume, Terminate, Abort, Show history, Description, Variables, Activity management, Delete finished, Delete selected, Check deadlines, Check limits.
F6.3	Flexibility to workflow participants	Access rights: various access rights – from administrator, projectmanager, moderator with full rights, to simple user - worker, player, author without administrative rights.
		Supports application mapping (map a package and package's processes applications to the real applications handled by a tool agent. Currently, six tool agents come with the Shark distribution), cache management (handle the size of the shark caches using this section).

Functional Auxiliary Functions

FA1 Statistics

Fct no	Function name	Function Implementation
FA1.1	Number of users	Support
FA1.2	Number of processes	Support
FA1.3	User rights	Support
FA1.4	Process Information	Support

FA2 HELP Functionality

Help functionalty is not provided by the administration or toolagent tools, but an understandable guide comes with the Shark installation package.

II Integral Characteristics

	Enhydra Shark				
Characteristics ISO/IEC 9126	**weak**	**good**	**strong**	**can't assess**	**score result**
Functionality		60%	40%		**4,8**
Reliability	25%	50%	25%		**4**
Usability	37,5%	62,5%			**3,25**
Efficiency	67%	33%			**2,66**
Maintainability		60%	40%		**4**
Portability	25%	60%	25%		**4,4**

III Pie - Charts

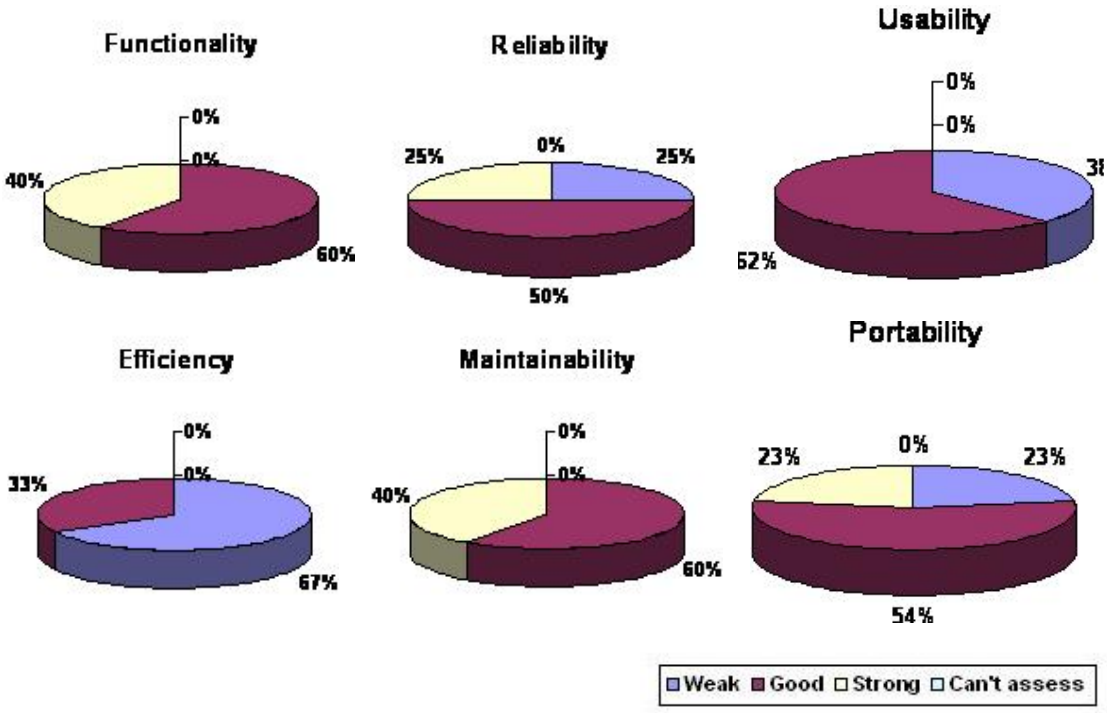


Figure 6.9 Pie-charts for Enhydra Shark's characteristics

EnterpriseXtention

I Functionality

Principal Functions

F1 Modelling, process definition

Fct no	Function name	Function implementation
F1.1	Process chart: definition, process thread	Process chart is transformed into the process model via graphic tools which produce XML (XPDL)
F1.2	Process: description, major, minor	Process definitions (models) are arranged into packages (according to XPDL specification) Each process definition includes the set of activities,

Fct no	Function name	Function implementation
		transitions and participants, specified as [v. 2.52] Lotus Notes documents with possibilities to export/import XPDL files; [v. 3.0] XML database will be used as the optional process storage. Process activities (of Generic, Complex, Route or Subprocess types) include “activity beans” definitions, split/join conditions and related transitions. Process participants/activity performers definitions are based on Person / Hierarchical Role / Functional Role / Organization Unit / System / Unknown (selected manually) / Relative (like activity #n performer or current performer’s manager) types. In every case, the engine resolves activity performers to real people or their proxies.
F1.3	Event: initial, result, triggering process thread, event result correlation	-
F1.4	Flowline	-
F1.5	Definition of organizational units	Organization units and Hierarchical / Functional roles are specified as workflow participants, the same way the people are. The real performer’s resolution is accomplished in the runtime, according to the Organization model and personal availability.
F1.6	Junctions	Flow controls (conditions) are specified in the process definition (model), based on events or current activity results. Manual controls are available as well.

F2 Simulation, debug
Not applicable

F3 Execution, workflow engine

F3.1	Processes: Deployment, initialization, repositories	Organization model creation by means of Java Organization Editor (JOE) graphicToXML tool and people assignment to the roles. Process model creation (or import from xpdl file) in the Process model database. Process validation (and simulation) All the elements of both models can be changed in the runtime without server restart.
F3.2	Management of distributed workflows	[v. 3.0] Distributed workflow support supported [v. 2.52] Distributed workflow possible for Lotus Notes users only. There are some limitations, for example in synchronisation between distributed workflows.
F3.3	Application Server Compatibility	[v. 3.0] Every WAS supporting J2EE may be used
F3.4	File formats	[v. 2.52] IBM Lotus Domino server installed, .nsf (notes

		storage fascility) files for the workflow engine [v. 3.0] As above OR Java files and .xml
F3.5	Transaction mechanisms	[v. 2.52] Uses native Lotus Notes mechanisms [v. 3.0] Own, built-in
F3.6	Supported communication protocol	NRPC, TCP, SMTP
F3.7	Users' roles	Role assignment in the Organization model database, system roles (ACL) are assigned by administrator by means of Administrative tool

F4 **Workflow client applications**

F4.1	User interface technologies: client application - type, proprietary (heavy weight user clients) or general purpose(light weight browsers)	[v. 2.52] For full functionality, including secure document access Lotus Notes client needed, reduced functionality available via MSIE browser [v. 3.0] Full functionality via lightweight browser

F5 Integration with other workflow engines; Supported standards

F5.1	Possibilities for collaboration with other workflow engines; standards	only XPDL

F6 Administration and monitoring

F6.1	Life cycle management	Monitoring of all processes available for administrator; built-in reporter; Process validation and simulation allows to Determine "dengerous time zone" and tune time schedule; in runtime administrator can browse for "dead" processes and manage time schedule and routing;
F6.2	Supervising capabilities. Process status.	Workflow log registers all process events. Workflow monitor changes statuses of the activities in the process according to the time schedule and process model. Built-in reporter shows the list of processes of different status (overall, per user or per process type)
F6.3	Flexibility to workflow participants	Flexible mechanism for automatic access rights assignment in the runtime; Manual access assignment available for administrators; X.Flow in the runtime resolves real persons or their proxies, according to the Organization model database. For B2B purpose, this database can store different organizations from different domains and can be used for execution of processes across the organizations;

Functional Auxiliary Functions

FA1 Statistics

Fct no	Function name	Function Implementation
FA1.1	Number of users	Tested for 4000 concurrent users on UNISYS 7000 (Intel) server

Fct no	Function name	Function Implementation
FA1.2	Number of processes	Tested to run 5000 processes daily
FA1.3	User rights	Flexible mechanism of runtime access rights assignment + manual rights assignment by administrators
FA1.4	Process Information	Graphic process viewer (Gantt style) TRIPLAN 1.2 Process progress visualized according to user's place in the Organization hierarchy or process ownership

FA2 HELP Functionalities
Help database & printed (or .pdf) manual

II Integral Characteristics

	enterpriseXtention				
Characteristics ISO/IEC 9126	**weak**	**good**	**strong**	**can't assess**	**score result**
Functionality		40%	60%		**5,2**
Reliability			75%	25%	**4,5**
Usability	25%	62.5%	12.5%		**3,75**
Efficiency		67%		33%	**2,68**
Maintainability		40%	20%	40%	**2,8**
Portability	40%	20%	20%	20%	**2,8**

III Pie Charts

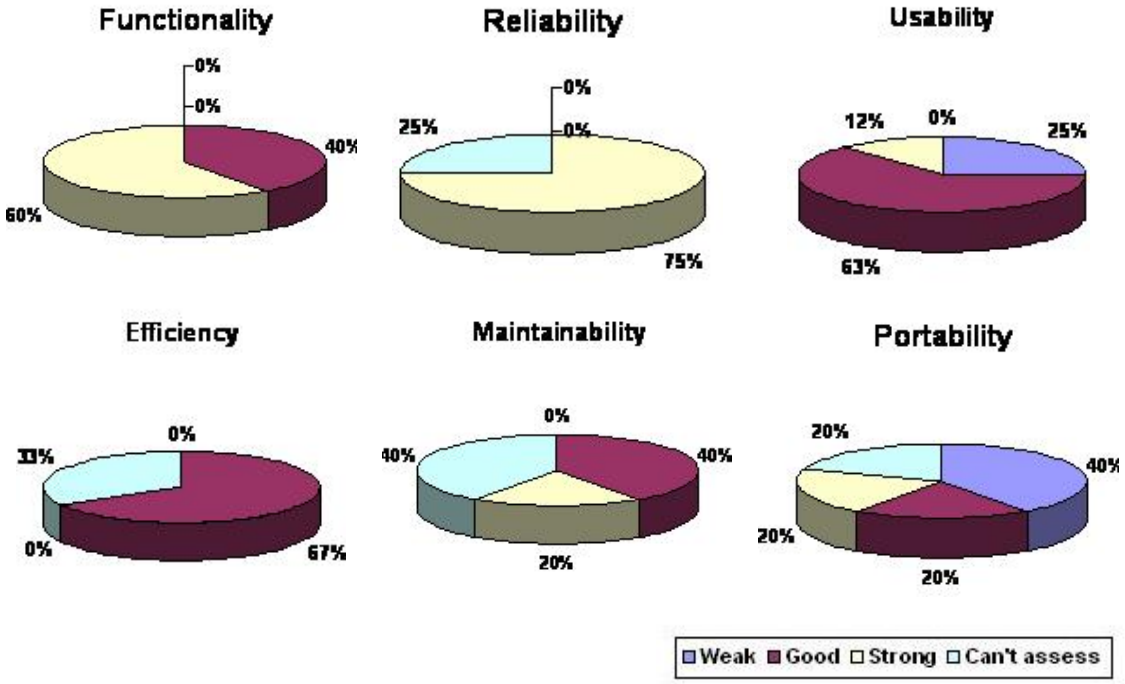


Figure 6.10 Pie-charts for EnterpricseXtention's characteristics

Freefluo

I Functionality

Principal Functions

F1 Modelling, process definition
Using Taverna's Workbench

F2 Simulation, debug

Fct no	Function name	Function implementation
F2.1	Objects: definition, simulation arrival profile, access data, simulation process	The simulation can be done only using the Taverna workflow system. The Taverna Workbench allows users to construct complex simulations from components located on both remote and local machines, run these workflows on their own data and visualize the results.
F2.2	Process: initialization, service time profile, necessary resources , available resources, child	Not applicable
F2.3	Event: description, simulation event (generation, definition of object, simulation arrival	Not applicable
F2.4	Junction: description (and, or, x or), symbol, simulation outflow	Not applicable
F2.5	Links: simulation probability, type	Not applicable
F2.6	General conditions: simulation validation, run simulation	Not applicable

F3 Execution, workflow engine

F3.1	Processes: Deployment, initialization, repositories	The workflow SCUFL file must be passed to engine via SOAP. In fact, the engine exposes some methods to receive as input SCUFL files.
F3.2	Management of distributed workflows	Not applicable
F3.3	Application Server Compatibility	All application servers, which are compatible with Apache-Tomcat-5.x
F3.4	File formats	XScufl files (XML format).
F3.5	Transaction mechanisms	Scufl specifications
F3.6	Supported communication protocol	SOAP and ASAP communication protocols are fully supported.
F3.7	Users' roles	Scufl specifications

F4 Workflow client applications

F4.1	User interface technologies: client application - type, proprietary(heavy weight user clients) or general purpose (light weight browsers)	The workflow engine operates with client applications using web services communication.

F5 Integration with other workflow engines; Supported standards
Not applicable

F6 Administration and monitoring

F6.1	Life cycle management	Only by Taverna workbench.
F6.2	Supervising capabilities Process status.	See above.
F6.3	Flexibility to workflow participants	J2EE web container security can be used to help secure a Freefluo web service.

Functional Auxiliary Functions

FA1 Statistics
Only by Taverna workbench

FA4 HELP Functionalities
Not applicable

II Integral Characteristics

	FreeFluo				
Characteristics ISO/IEC 9126	**weak**	**good**	**strong**	**can't assess**	**score result**
Functionality	40%	60%			**3,2**
Reliability	25%	25%		50%	**1,5**
Usability	75%			25%	**1,5**
Efficiency		67%		33%	**2,68**
Maintainability	20%	60%		20%	**2,8**
Portability	40%	20%	40%		**4**

III Pie - Charts

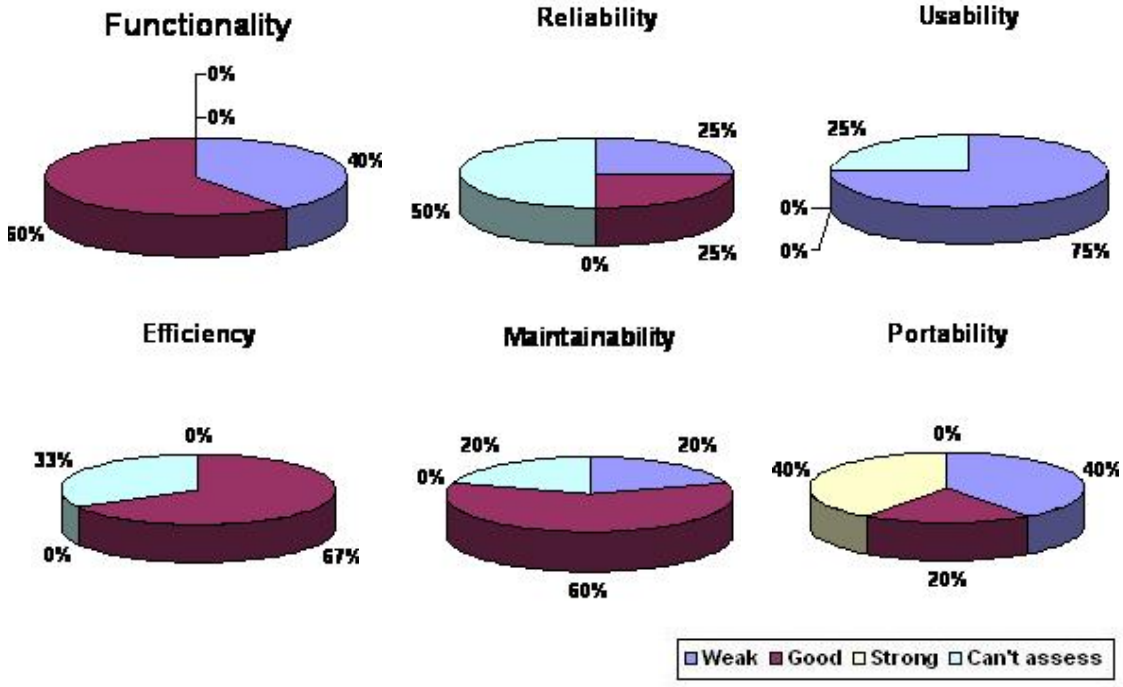


Figure 6.11 Pie-charts for Freefluo's characteristics

JBoss jBPM

I Functionality

Principal Functions

F1 Modelling, process definition

Fct no	**Function name**	**Function implementation**
F1.1	Process chart: definition, process thread	3 categories of objects : - states objects : Start state (where the process is beginning), End state (where the process is ending), State object (custom state) - nodes : fork (launch several tasks simultaneously), join (wait for all tasks to finish), decision (the process must take a decision depending on some data, like testing if account balance is positive for loan approval), task node (represents one or more tasks that are to be performed by humans), node (serves the situation where you want to write your own code in a node but is also responsible for propagating the execution) , state (the process goes into a wait state and waits for a message to resume its execution) - transitions: connect nodes Actions are JAVA classes implementing a given interface. They can be executed either on a task (and then is responsible for propagating the execution) or an event (entering/leaving a

Fct no	**Function name**	**Function implementation**
		node, taking a transition)
F1.2	Process: description, major, minor	Processes are called Actions. Actions implement the interface ActionHandler provided by jBpm. This interface defines only one method to be implemented : execute (EvoutionContext ec). This method is the core method called when the Action is to be executed. The parameter EvolutionContext provides access to the process execution context information like accessing variables. Processes can perform whatever you want, like updating a database. Actions can be performed at two different times: on an event (like entering a node) or on a node.
F1.3	Event: initial, result, triggering process thread, event result correlation	Events specify moments in the execution of the process. The jBPM engine will fire events during graph execution. This occurs when jbpm calculates the next state. An event is always relative to an element in the process definition like e.g. the process definition, a node or a transition. Most process elements can fire different types of events. A node for example can fire a node-enter event and a node-leave event. Events are the hooks for actions. Each event has a list of actions. When the jBPM engine fires an event, the list of actions is executed. External events can also occur: the reception of a message from an external system can resume the execution of the process diagram. However, everything must be done outside the process definition file.
F1.4	Flowline	Flowlines are Transitions. Transitions don't do anything, except to define a path through Nodes. It is possible to execute an Action when the transition is processed.
F1.5	Definition of organizational units	N/A
F1.6	Junctions	Junctions don't exist on their own. The only way is to add a Decision node. There are basically 2 ways to specify the decision criteria. Simplest is by adding condition elements on the transitions. Conditions are beanshell script expressions that return a boolean. At runtime the decision node will loop over its leaving transitions (in the order as specified in the xml), and evaluate each condition. The first transition for which the conditions resolves to 'true' will be taken. Alternatively, an implementation of the DecisionHandler can be specified. Then the decision is calculated in a java class and the selected leaving transition is returned by the decide-method of the DecisionHandler implementation. DecisionHandler interface defines a unique method: String decide (ExecutionContext ec). It must return the name of the transition to follow.

F2 Simulation, debug
Not applicable: jBPM does not come with a debugger tool or a simulation tool

F3 **Execution, workflow engine**

F3.1	Processes: Deployment, initialization, repositories	The deployment of the process diagram must be done according this: - the definition file (XML), must be packed in a .PAR archive (Process Archive). The deployment can be done in 3 different ways : via the Designer tool, an Ant Task or programmaticaly (both using ProcessArchive-DeployerTask Class. Deploying a jBPM process is saving its definition in a jBPM database (jBPM supports Oracle, MySQL, MSQL…). It also saves the O/R model used for Hibernate
F3.2	Management of distributed workflows	N/A
F3.3	Application Server Compatibility	jBPM is a JAVA component that can be used in any J2SE platform, and thus in any Web application server
F3.4	File formats	A PAR archive contains the processdefinition.xml file is mandatory for the deployment, but then, only JAVA classes are needed.
F3.5	Transaction mechanisms	Transactions are supported in the jBPM API (beginTransaction(), commitTransaction() and rollbackTransaction() called on a jBpmSession object. The behaviour of these methods depends on the Hibernate configuration : If you have configured hibernate to manage its own JDBC connections, the transaction operations will result in the corresponding operations on the JDBC connection. Otherwise the transaction methods will call the corresponding methods on the UserTransaction object in the container.
F3.6	Supported communication protocol	As said before, all kind of communication protocols can be used as long as the corresponding libraries are accessible along your application. For instance you can easily add a SOAP/HTTP support to your application if you deploy it on an J2EE application server.
F3.7	Users' roles	User roles are called Swimlanes. It specifies that multiple tasks in the process should be done by the same actor.

F4 Workflow client applications

F4.1	User interface technologies: client application - type, proprietary(heavy weight user clients) or general purpose(light weight browsers)	You can have any kind of client you want as long as the application is deployed on a suitable platform. For instance you can interact with the application with a HTML browser if your application is available on a Web Application Server. You can also use heavyweight clients with Swing, or whatever client you want as long as your application as the appropriate connector to interact with.

F5 Integration with other workflow engines; Supported standards

F5.1	Possibilities for collaboration with other workflow engines; standards	jBPM has been developed to run jPDL process charts. Though it is possible to run BEPL process diagrams. That is possible by adding a plugin. It is important to say that BEPL will be soon executed in native in the upcoming releases of jBPM.

F6 Administration and monitoring

F6.1	Life cycle management	Time scheduling is available via a Scheduler component provided. A timer that is specified on a node, is not executed after the node is left. Both the transition and the action are optional. When a timer timer is executed, the following events occur in sequence : • an event is fired of type timer • if an action is specified, the action is executed. • if a transition is specified, a signal will be sent to resume execution over the given transition. Every timer must have a unique name. Example of timer definition : <timer name='reminder' duedate='3 business hours' repeat='10 business minutes' transition='time-out-transition' > <action class='the-remainder-action-class-name' /> </timer> In this example the Action class will be executed every 10 business minutes and for 3 business hours. Note that this component comes with a specific notation : <quantity> [business] <unit> Where <quantity> is a piece of text that is parsable with Double.parseDouble(quantity). <unit> is one of {second, seconds, minute, minutes, hour, hours, day, days, week, weeks, month, months, year, years}. And adding the optional indication business means that only business hours should be taken into account for this duration. Without the indication business, the duration will be interpreted as an absolute time period. A timer on a task will be cancelled when the task is ended (=completed). However, it can be changed by using the cancel-event attribute to task-assign or task-start.
F6.2	Supervising capabilities. Process status.	jBPM comes with a administration web application.
F6.3	Flexibility to workflow participants	Access rights

F7 Other

Exception handling The exception handling mechanism of jBPM only applies to java exceptions. Graph execution on itself cannot result in problems. It is only the execution of delegation classes that can lead to exceptions.

On process-definitions, nodes and transitions, a list of exception-handlers can be specified. Each exception-handler has a list of actions. When an exception occurs in a delegation class, the process element parent hierarchy is serached for an appropriate exception-handler. When it is found, the actions of the exception-handler are executed.

Note that the exception handling mechanism of jBPM is not completely similar to the java exception handling. In java, a caught exception can have an influence on the control flow. In the case of jBPM, control flow cannot be changed by the jBPM exception handling mechanism. The exception is either caught or uncaught. Uncaught exceptions are thrown to the client or the exception is caught by a jBPM exception-handler. For caught exceptions, the graph execution continues as if no exception has occurred.

Functional Auxiliary Functions

FA1 Statistics
Not applicable

FA2 HELP Functionalities
Help available online on the jBoss website

II Integral Characteristics

	jBPM				
Characteristics ISO/IEC 9126	**weak**	**good**	**strong**	**can't assess**	**score result**
Functionality	20%	60%	20%		**4**
Reliability	25%	50%		25%	**2,5**
Usability	37,5%	37,5%		25%	**2,25**
Efficiency		50%		50%	**2**
Maintainability	20%	20%	40%	20%	**3,6**
Portability	20%	20%	40%	20%	**3,6**

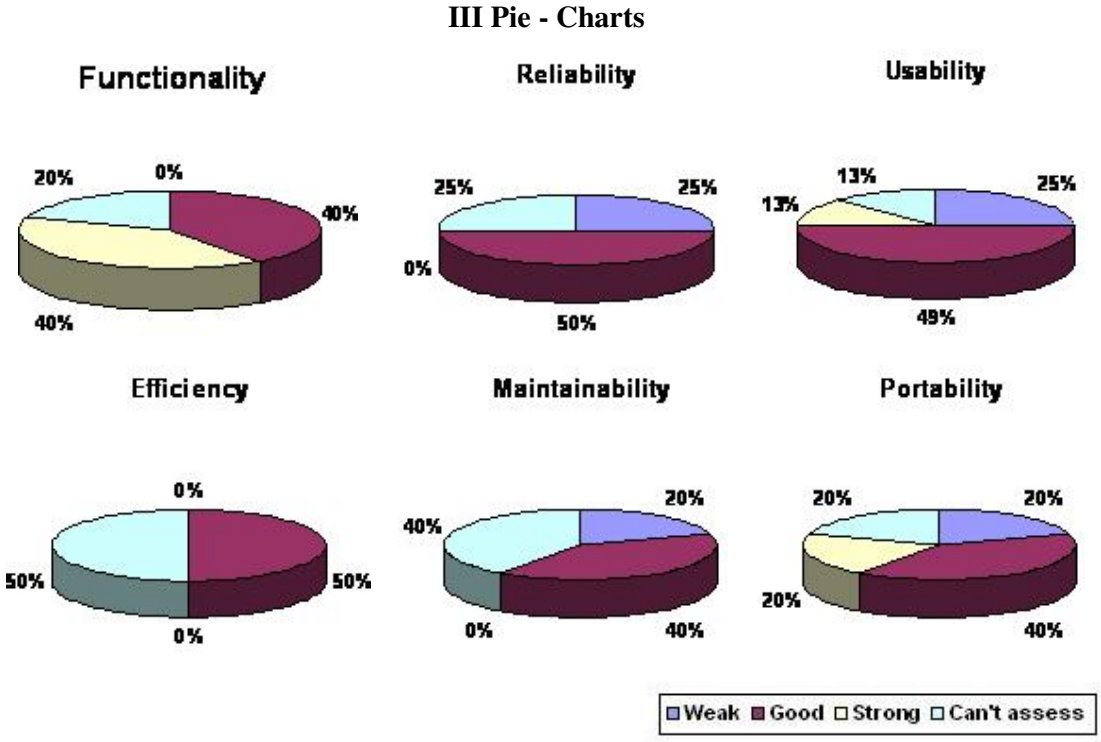


Figure 6.12 Pie-charts for JBossjBPM's characteristics

ObjectWeb BONITA

I. Functionality

Principal Functions

F1 Modelling, process definition

Fct no	Function name	Function implementation
F1.1	Process chart: definition, process thread	Terminology in BONITA - A **process** is a set of activities. In BONITA, the term project is also used. - An **activity** is an atomic unit of work. In BONITA, activities are also termed **Nodes**. - A **transition** is a dependency expressing an order constraint between two activities. In BONITA, **transitions** are also termed **Edges**. - A **property** is a workflow unit of data, commonly known as workflow relevant data. - A **hook** is a user-defined logic adding automatic behaviour to activities/nodes. - A **mapper** is a unit of work allowing dynamically roles resolution at workflow instantiation time. - A **performer assignment** is a unit of work adding additional activity assignment rules at run time.
F1.2	Process:	**Process**

Fct no	Function name	Function implementation
	description, major, minor	Bonita supports both cooperative and administrative workflows processes. These processes are mapped on three Bonita types: - Cooperative: flexible workflow process allowing definition and execution operations just after the process is created; - Model: workflow process, which contains the workflow definition logic; These projects can be instantiated by users; - Instance: workflow process representing a specific execution of a workflow model; The status of a workflow process can be controlled at definition or runtime by the workflow process administrator/s. Two possible statuses are allowed for a workflow process: - Active: the workflow process could be modified or executed. This is the default status for a cooperative, model or instance process. - Hidden: the process is not yet available. Operations like execution, cancel or termination of cooperatives and instances projects are not allowed as well as model instantiation. This is the status mode which allows models modifications after those ones were instantiated. **Life Cycle of Process** BONITA has a very simple process life cycle - A process is initial one it has been created. As soon as the process is in this state, it can be controlled using both User API & Project API. The User API allows monitoring the execution of the process. Whenever the first activity has been started using the User API, the process goes to started state. The execution of the process is performed by the BONITA enactment engine, under control of applications with the User API. - A process is started as soon as the first activity has started. While being executing, the process definition still can be modified using the Project API. When all activities are terminated, the process stays in state started. It still can be modified, for example, new activities can be added. - A process is terminated once it has been explicitly terminated by an application thru the User API. In terminated state, the process definition cannot be modified any more.
F1.3	Event: initial, result, triggering process thread, event result correlation **Or for F1.5**	Event – in BONITA called **Activity** The activity is the basic unit of work within a process. Execution of an activity can either be automatic, or manual : - **Automatic**: In this case, the BONITA enactment engine will start it as soon as the applicable transitions from preceding activities are successfully evaluated. - **Manual**: the BONITA enactment engine will not start a manual activity until some application has explicitly started it through the User API. Any **activity** is associated with a **role**. All the users being allocated that role in the scope of the process have the possibility to take over the activity. Any **activity** is enclosed in a **Transaction**, and every call to a method

Fct no	Function name	Function implementation
		of the Bonita API, which changes the state of an activity, is considered as part of a transaction (that means every method but those beginning with "getxxx" which only retrieve information). Concept of **Hooks** **Hooks** are user-defined logic that can be triggered at some defined points in the life of the activity. Those defined points are : - **Before Start hook** is called just before the activity starts. The Before Start hook is not considered to be in the same transaction than the activity. The Before Start hook is not triggered for automatic activities. - **After Start hook** is called just after the activity has started. It is considered to be in the same transaction than the activity. The After Start hook is not triggered for automatic activities that cannot be anticipated. - **Cancel hook** is called before cancelling an activity and it's considered to be in the same transaction than the activity. - **Before Terminate hook** is called just before the activity terminates. The Before Terminate hook is considered to be in the same transaction than the activity. - **After Terminate hook** is called just after the activity has terminated. It is not considered to be in the same transaction than the activity. - **Anticipating hook** is called when an automatic activity is started, only if the activity is anticipable. It is considered to be in the same transaction than the activity. - **OnReady hook** is called when an activity becomes ready, so it would be veiy useful to notify some kind of information to the user responsible to execute it. It is not considered to be in the same transaction than the activity. - **On Deadline hook** is called when the activity deadline expires. It is not considered to be in the same transaction than the activity.
F1.4	Flowline	-
F1.5	Definition of organizational units	**See F1.3**
F1.6	Junctions	**Transition between activities:** Most of the usual transition patterns can be achieved with BONIT A. There is no special node to achieve these patterns; rather any activity can act as a routing node. The transition pattern will be determined according to the type of the activity, which can be one of AND-JOIN (also known as "synchronize"), or OR-JOIN (also known as " asynchronous join"). The transition pattern will also be determined from and the number of outgoing edges from an activity; that is the SPLIT construct, (which allows having several activities executing in parallel) is not a specific type of activities; if there are several outgoing nodes from a given activity, then it is a SPLIT construct.

F2 Simulation, debug
Not available yet. Work in progress.

F3 Execution, workflow engine

F3.1	Processes: Deployment, initialization, repositories	
F3.2	Management of distributed workflows	
F3.3	Application Server Compatibility	Yes **JOnAS server**: JOnAS is a pure Java, open source, application server, developed within the ObjectWeb consortium. Its high modularity allows to it to be used as: - a J2EE server, for deploying and running EAR applications (i.e. applications composed of both web and ejb components), - an EJB container, for deploying and running EJB components (e.g. for applications without web interfaces or when using JSP/Servlet engines that are not integrated as a JOnAS container), - a Web container, for deploying and running JSPs and Servlets (e.g. for applications without EJB components).
F3.4	File formats	
F3.5	Transaction mechanisms	**See F1.6**
F3.6	Supported communication protocol	SOAP and XML - The Bonita Workflow also includes a browser-based environment with Web Services integration using SOAP and XML.
F3.7	Users' roles	Users - BONITA make the distinction between **Users** and **Participants**: - Users are people who will make use of the workflow system (whatever process they will be part of). - Participants are all the users that are allowed to play some role in a given process. First, the user has to be registered in Bonita System, for authentication need (using Bonita User Registration API). Then, he has to be declared as participant in each project, in which he is involved in (using Bonita Project API). He is then able to take part in the process. Users are managed in BONITA specific base (or thru a LDAP repository as well). This base allows storing properties (also called preferences) for a given user. Properties are (key, value) pairs where both key and values are String variables. The application can set and retrieve properties using the User interface. BONITA makes use of specific user properties in order to store the User preferences. **Relationship to processes** - users have to be explicitly associated to processes in order to participate and to have

		visibility of events occurring in those processes. Two scenarios allow to associate a User with a process - that is make a User a Participant of this process: - Whenever a process is created, it is created on behalf of the User that initiated the Project Interface. This user is automatically associated to the newly created process, and can from now on assume the Admin role in the scope of the process. - The users assuming the admin role for a given process has the right to associate new users to the process, and to allocate any role to them.

F4 Workflow client applications

F4.1	User interface technologies: client application - type, proprietary (heavy weight user clients) or general purpose (light weight browsers)	Yes Bonita system have built in client applications developed by ECOO team www.inria.fr/recherche/equipes/ecoo.fr.html

F5 Integration with other workflow engines; Supported standards

F5.1	Possibilities for collaboration with other workflow engines; standards	Not available yet Work in progress.Possibilities for collaboration with other XPDL based workflow engines: XPDL module supporting import/export XML files conforming to XPDL specification. This service will be integrated in GraphEditor application.

F6 Administration and monitoring

F6.1	Life cycle management	**Life Cycle of Process** BONITA has a very simple process life cycle - A process as initial once it has been created. As soon as the process is in this state, it can be controlled using both User API & Project API. The User API allows monitoring the execution of the process. Whenever the first activity has been started using the User API, the process goes to started state. The execution of the process is performed by the BONITA enactment engine, under control of applications with the User API. - A process is started as soon as the first activity has started. While being executing, the process definition still can be modified using the Project API. When all activities are terminated, the process stays in state started. It still can be modified, for example, new activities can be added. - A process is terminated once it has been explicitly terminated by an application thru the User API. In terminated state, the process definition cannot be modified any more.
F6.2	Supervising capabilities Process status	
F6.3	Flexibility to workflow participants	Users - BONITA make the distinction between **Users** and **Participants**: - Users are people who will make use of the workflow system

		(whatever process they will be part of). - Participants are all the users that are allowed to play some role in a given process.

Functional Auxiliary Functions

FA1 Statistics
Not applicable

FA2 HELP Functionalities
Not applicable

II Integral Characteristics

	ObjectWeb BONITA				
Characteristics ISO/IEC 9126	**weak**	**good**	**strong**	**can't assess**	**score result**
Functionality		60%		40%	**2,4**
Reliability	25%	50%		25%	**2,5**
Usability		50%		50%	**2**
Efficiency		67%		33%	**2,68**
Maintainability		80%		20%	**3,2**
Portability		40%		60%	**1,6**

III Pie - Charts

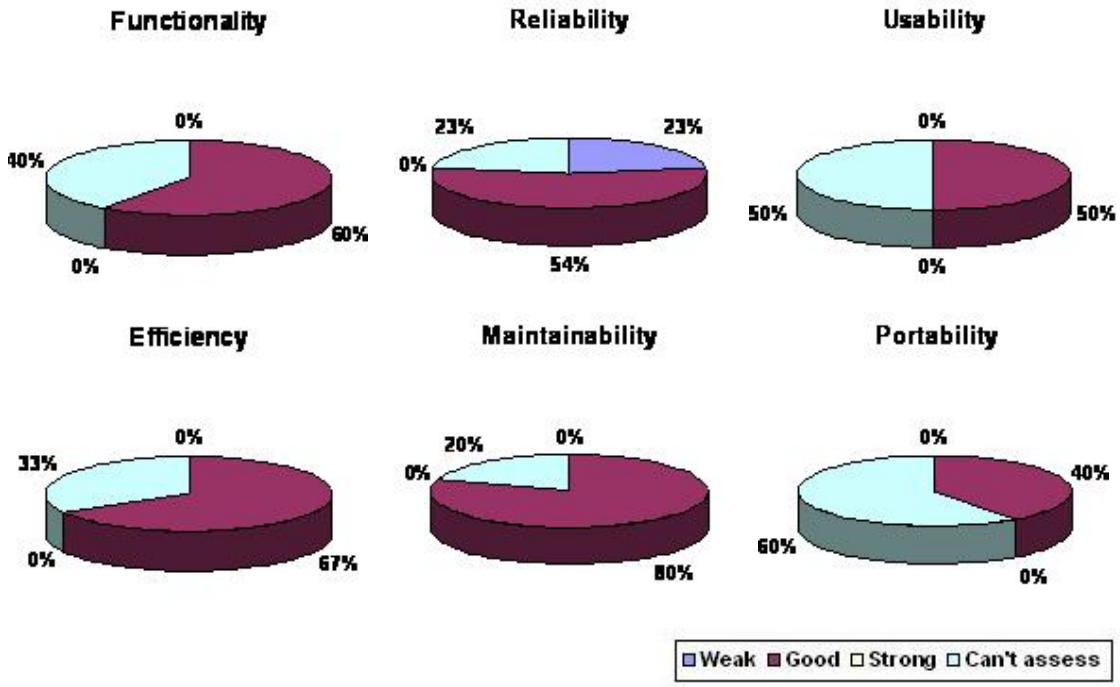


Figure 6.13 Pie-charts for BONITA's characteristics

OpenWFE

I. Functionality

Principal Functions

F1 Modelling, process definition

Fct no	Function name	Function implementation
F1.1	Process chart: definition, process thread	OpenWFE has its own Process Definition Language stored as XML structure. OpenWFE gives possibility to define business process in three manner: • directly by editing XML files, • using visual builder through web browser (Droflo application), • using visual builder running as Eclipse plug-in
F1.2	Process: description, major, minor	Each process in OpenWFE is defined in different XML file in stored in *workflow-definitions* dir. It is possible to define processes and subprocesses using process-*definition* and *subprocess* tags. It is possible to include definitions of processes from other files. To define the whole business process, a designer has to: • define participants • create workflow definition describing the flow of the information between participants, • create stores, usually one per participant, • create users and give them rights to launch the flow, edit parameters, proceed and delegate workitems
F1.3	Event: initial, result, triggering process thread, event result correlation	
F1.4	Flowline	Users can monitor the current state of the workflow using webclient application.
F1.5	Definition of organizational units	Organizational units are not represented in OpenWFE. It's possible to set up the organizational unit as a participant in a workflow
F1.6	Junctions	

F2 Simulation, debug

Process definitions can be checked by the validate-flowdef script before running the engine. OpenWFE starts in console so all errors are displayed. All events are also logged to dedicated files separatelly for engine, worklist, webclient, etc.

F3 Execution, workflow engine

F3.1	Processes: Deployment, initialization, repositories	Not applicable All changes in the workflow are done in config files. To apply changes the server has to be restarted.
F3.2	Management of distributed workflows	OpemnWFE supports distributed workflows. The engine uses dispatcher and listeners that uses files, TCP socket, and mail boxes. It's possible to implement them in other

		way
F3.3	Application Server Compatibility	OpenWFE includes webclient which is web application written in Java. The Jetty (java webserver) engine is included together in the OpenWFE package application. But the end user can switch to another webserver for example
F3.4	File formats	OpenWFE for configuration and operation (fe storing expressions and workitems in the filesystem or for dispatching a workitem to a participant) uses xml files encoded ISO-8859-1. Implementation of persistence in DB and in memory is included in OpenWFE package. The user can choose between them.
F3.5	Transaction mechanisms	The OpenWFE has build in transaction mechanism. It is supported by *do*, *undo* and *redo* tags in workflow definition.
F3.6	Supported communication protocol	TCP, SMTP/POP3/IMAP
F3.7	Users' roles	User's are connected to participants (le in a workflow). One user can play many roles.

F4 Workflow client applications

F4.1	User interface technologies: client application - type, proprietary(heavy weight user clients) or general purpose(light weight browsers)	OpenWFE includes webclient which is web application written in Java. The Jetty (java webserver) engine is included together in the OpenWFE package application. But the end user can switch to another webserver for example

F5 **Integration with other workflow engines; Supported standards**

F5.1	Possibilities for collaboration with other workflow engines; standards	Not supported

F6 Administration and monitoring

F6.1	Life cycle management	OpenWFE has built-in cronService. The service is configured in workflow definitions.
F6.2	Supervising capabilities. Process status.	Through the control interface administrator can manage all workflows and their participants. There is also command line tool that can be used to admin the engine
F6.3	Flexibility to workflow participants	One user can play some roles in a workflow.

Functional Auxiliary Functions

FA1 Statistics
Not applicable

FA2 HELP Functionalities
Not applicable

II Integral Characteristics

	OpenWFE				
Characteristics ISO/IEC 9126	**weak**	**good**	**strong**	**can't assess**	**score result**
Functionality	20%	20%	60%		
Reliability		25%	25%	50%	
Usability	12.5%	50%	25%	12.5%	
Efficiency		33%	67%		**5,34**
Maintainability			100%		**6**
Portability	20%		80%		

III Pie - Charts

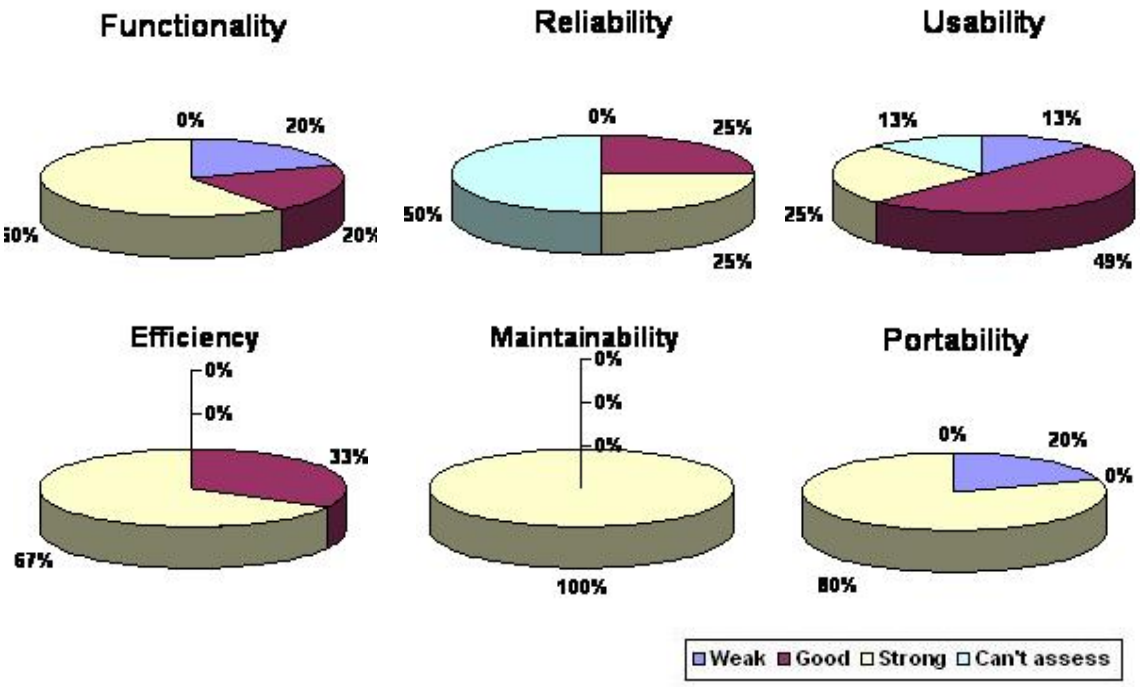


Figure 6.14 Pie-charts for OpenWFE's characteristics

Oracle BPEL Process Manager

I. Functionality

Principal Functions

F1 Modelling, process definition

F1.1	Process chart: definition, process thread	*Oracle BPEL Process Manager* fully supports BPEL 1.1. Additional concepts concerning the integration of human users are explained in section 1.1.4. One consequence of the BPEL 1.1
F1.2	Process: description, major, minor	

F1.3	Event: initial, result, triggering process thread, event result correlation	specification is that it is possible to invoke one BPEL process from another BPEL process utilizing the corresponding WSDL interface. The concept of sub-processes is not supported in BPEL 1.1. *BPEL Designer* is a graphical tool for the definition of BPEL processes. The current version is based on various wizards that ease the task of defining consistent workflow models a lot. In practise, it is hard to define complex workflows utilizing graphical tools because the graphical notation is not sufficiently compact and unclear in details. The same is true for the lengthy XML representation of the BPEL workflow.
F1.4	Flowline	
F1.5	Definition of organizational units	
F1.6	Junctions	

F2 Simulation, debug

F2.1	Objects: definition, simulation arrival profile, access data, simulation process	The *BPEL Designer* is able to validate the workflow definition either on demand or as a prerequisite of the compilation and deployment process. The *BPEL Console* supports debugging, flow visualization, and monitoring of BPEL instances.
F2.2	Process: initialization, service time profile, necessary resources , available resources, child diagram	
F2.3	Event: description, simulation event (generation, definition of object, simulation arrival profile)	
F2.4	Junction: description (and, or, x or), symbol, simulation outflow	
F2.5	Links: simulation probability, type	
F2.6	General conditions: simulation validation, run simulation	

F3 Execution, workflow engine

F3.1	Processes: Deployment, initialization, repositories	The *BPEL Designer* creates a .jar archive from the BPEL and WSDL specification of the workflow
F3.2	Management of distributed workflows	A BPEL workflow can invoke other (BPEL) workflows utilizing their WSDL interface. Distributed transactions are not supported.
F3.3	Application Server Compatibility	*BPEL Server* runs either within JBoss or WebLogic application servers. The exhaustive version runs only within Oracle AS.
F3.4	File formats	The workflow itself is compiled into Java code. BPEL and WSDL descriptions are included in the .jar archive. *BPEL Console* administers several workflow specific attributes.
F3.5	Transaction mechanisms	BPEL Server supports the BPEL 1.1 transaction mechanisms. Utilizing the proprietary Java

		Business Delegate Mechanism BPEL workflows can be integrated into distributed JTA transactions.
F3.6	Supported communication protocol	*BPEL Server* communicates via SOAP considering WS-Addressing.
F3.7	Users' roles	*BPEL Process Manager* supports an identity service that runs on top of various user management infrastructures like for example LDAP. The arrangement of users to groups/roles is supported and utilized in the workflow services.

F4 Workflow client applications

F4.1	User interface technologies: client application - type, proprietary(heavy weight user clients) or general purpose(light weight browsers)	Workflow clients can communicate with BPEL processes either via SOAP or via the proprietary Java *Business Delegate Mechanism.* Only communication utilizing Axis 1.1 has been evaluated.

F5 Integration with other workflow engines; Supported standards

F5.1	Possibilities for collaboration with other workflow engines; standards	In principle is it possible to communicate with every engine that supports SOAP and WS-Addressing.

F6 Administration and monitoring

F6.1	Life cycle management	The *BPEL Console* supports various mechanisms to administer BPEL processes and instances utilizing proprietary interfaces.
F6.2	Supervising capabilities Process status	
F6.3	Flexibility to workflow participants	

F7 Deployment and Binding

Deployment of BPEL processes can be triggered by the *BPEL Designer* as well as by the *BPEL Console*. Binding to Web services endpoints is done via the information provided in the WSDL. It is foreseen to support the modification of Web service endpoints or the definition of multiple endpoints in the *BPEL Console*.

Functional Auxiliary Functions

FA1 Statistics

Not applicable

FA2 HELP Functionalities

Oracle provides comprehensive documentation and examples for the *BPEL Process Manager*.

II Integral Characteristics

	Oracle BPEL Process Manager				
Characteristics ISO/IEC 9126	**weak**	**good**	**strong**	**can't assess**	**score result**
Functionality		40%	40%	20%	**4**
Reliability		75%		25%	**3**
Usability		75%	25%		**4,5**
Efficiency		67%		33%	**2,68**
Maintainability		20%		80%	**0,8**
Portability		60%	20%	20%	**3,6**

III Pie - Charts

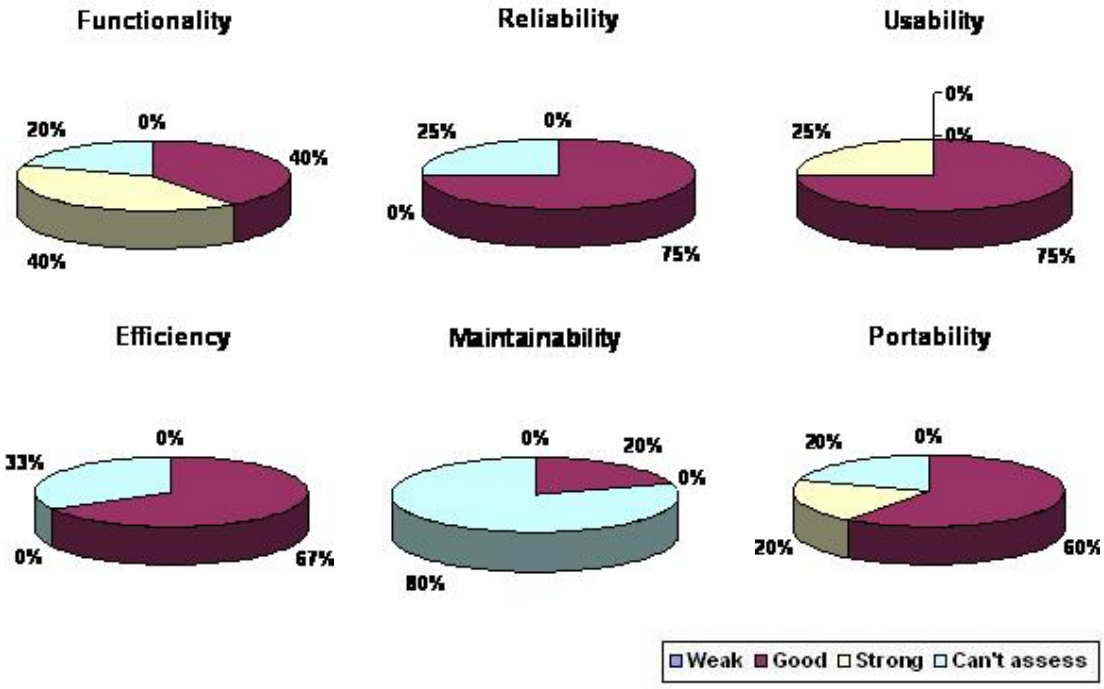


Figure 6.15 Pie-charts for Oracle BPEL Process Manager's characteristics

Process Modeler

I. Functionality

Principal Functions

F1 Modelling, process definition

Fct no	Function name	Function implementation
F1.1	Process chart: definition, process thread	*Process Modeler* for Microsoft Visio is a user-friendly graphical modelling editor. The business process diagram

Fct no	Function name	Function implementation
		tools palette in the editor provides the upper basic elements for designing a BPMN diagram. The user can drag and drop elements to a BPMN diagram and set after that their properties in a special window.
F1.2	Process: description, major, minor	not applicable
F1.3	Event: initial, result, triggering process thread, event result correlation	In BPMN, *events* are Start events, End events, and Intermediate events. Processes are Tasks or Sub-Processes (expand or collapsed); *Process Modeler* contains these elements.
F1.4	Flowline	*Process Modeler* fully supports BPMN 1.0. So every kind of Flowline-modelling is possible if it is possible in BPMN.
F1.5	Definition of organizational units	In BPMN, organizational units are Groups, Pools or Lanes. *Process Modeler* contains these elements.
F1.6	Junctions	*Process Modeler* fully supports BPMN 1.0. So every kind of junction is possible if it is possible in BPMN.

F2 Simulation, debug
It is not applicable (BPMN is not an executable format)

F3 Execution, workflow engine
It is not applicable (BPMN is not an executable format)

F4 Workflow client applications
It is not applicable (BPMN is not an executable format)

F5 Integration with other workflow engines; Supported standards
It is not applicable

F6 Administration and monitoring
It is not applicable (BPMN is not an executable format)

Functional Auxiliary Functions

FA1 Statistics

FA1.1	Number of users	One user
FA1.2	Number of processes	not applicable
FA1.3	User rights	not applicable
FA1.4	Process Information	*Process Modeler* allows exporting out of the model information into the Microsoft Word environment.

FA2 HELP Functionalities
Validation Function

II Integral Characteristics

	Process Modeler				
Characteristics ISO/IEC 9126	**weak**	**good**	**strong**	**can't assess**	**score result**
Functionality	20%	80%			**3,6**
Reliability		100%			**4**
Usability	25%	50%	25%		**4**
Efficiency		33%		67%	**1,32**
Maintainability	100%				**2**
Portability	60%		40%		**3,6**

III Pie - Charts

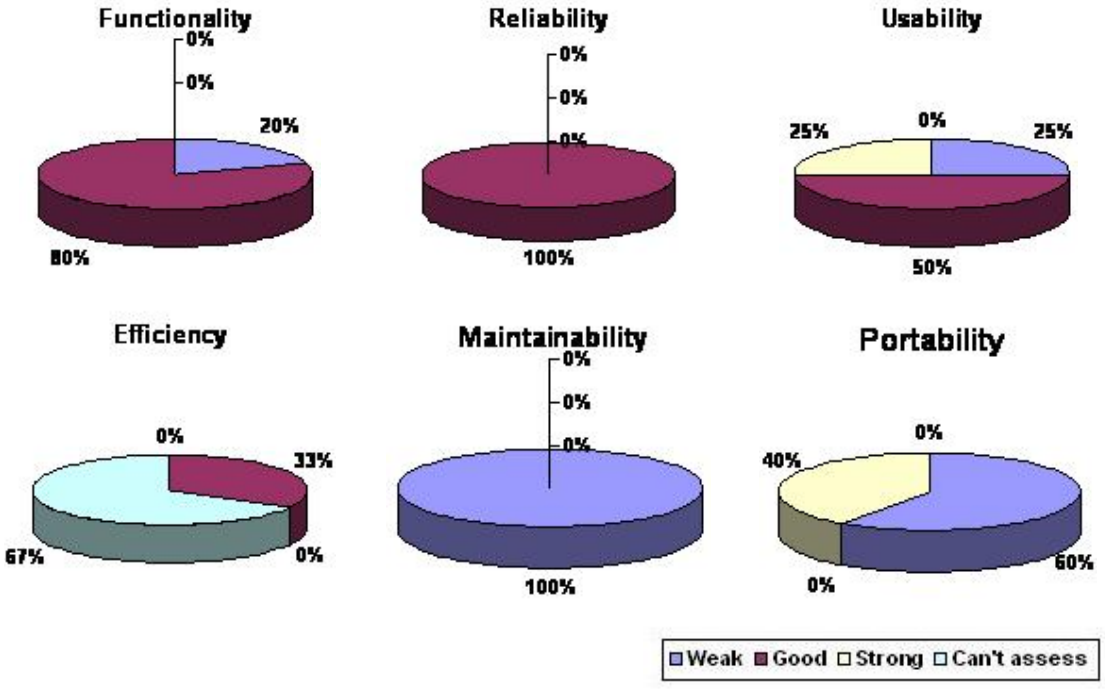


Figure 6.16 Pie-charts for Process Modeler for Microsoft Visio's characteristics

SAP NetWeaver Exchange Infrastructure

I. Functionality

Principal Functions

F1 Modelling, process definition

F1.1	Process chart: definition, process thread	
F1.2	Process: description, major, minor	

F1.3	Event: initial, result, triggering process thread, event result correlation	
F1.4	Flowline	
F1.5	Definition of organization units	
F1.6	Junctions	

F2 Simulation, debug

Not applicable

F3 Execution, workflow engine

F3.1	Processes: Deployment, initialization, repositories	Business Processes and Business Scenarios are stored in a repository called Integration Repository and brought to the Integration engine for execution.
F3.2	Management of distributed workflows	-
F3.3	Application Server Compatibility	N/A
F3.4	File formats	-
F3.5	Transaction mechanisms	Transactions are supported with BTP protocol.
F3.6	Supported communication protocol	Communication protocols supported are SOAP, HTTP(S), FTP, SMTP, FILE, JDBC, JMS
F3.7	Users' roles	-

F4 Workflow client applications

F4.1	User interface technologies: client application - type, proprietary(heavy weight user clients) or general purpose(light weight browsers)	Workflow clients can be of various types' grace of the great number of adapters available to communicate with external applications.

F5 Integration with other workflow engines; Supported standards

F5.1	Possibilities for collaboration with other workflow engines; standards	Workflow collaboration can be achieved with web services ends. It also supports WSCI for choreography workflow.

F6 Administration and monitoring

F6.1	Life cycle management	
F6.2	Supervising capabilities Process status	
F6.3	Flexibility to workflow participants	

II Integral Characteristics

	SAP NetWeaver Exchange Infrastructure				
Characteristics ISO/IEC 9126	**weak**	**good**	**strong**	**can't assess**	**score result**
Functionality		60%	20%	20%	**3,6**
Reliability	25%	50%		25%	**2,5**
Usability		75%		25%	**3**
Efficiency	33%	34%		33%	**2,02**
Maintainability	20%	60%		20%	**2,8**
Portability	20%	40%		40%	**2**

III Pie - Charts

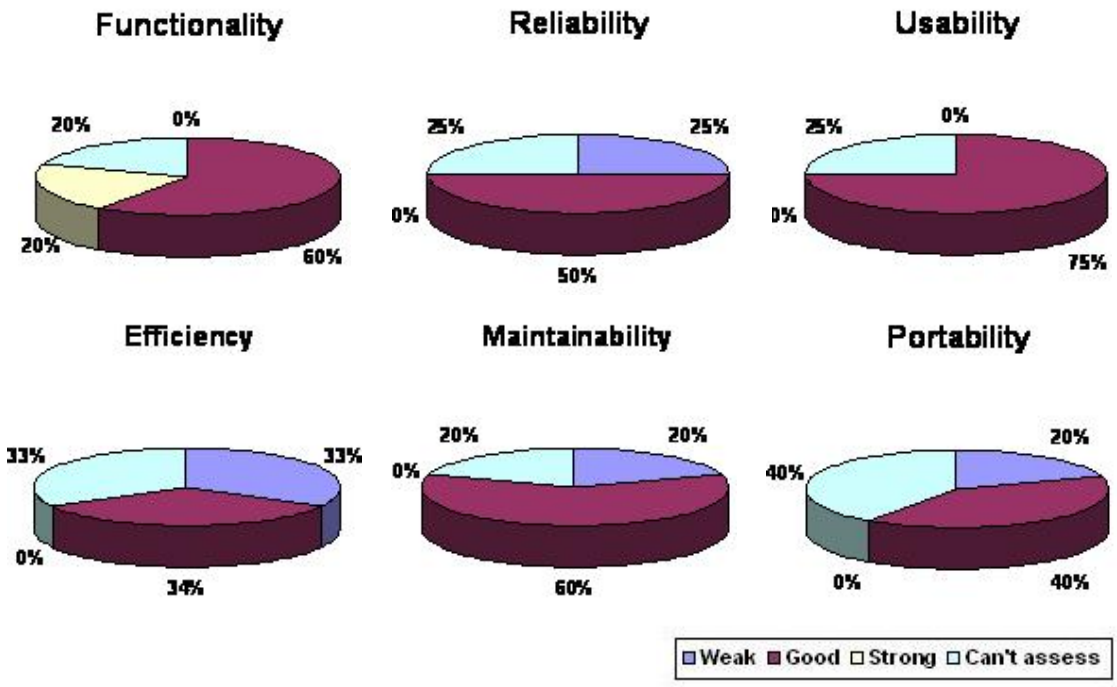


Figure 6.17 Pie-charts for SAP NetWeaver Exchange Infrastructure's characteristics

WfMOpen

I. Functionality

Principal Functions

F1 Modelling, process definition

Not applicable (the WfMOPEN only features the WF engine).

F2 Simulation, debug

F2.1 – F2.6 Not Applicable

F2.7 DEBUGGING: Workflows may be run in debugging mode. In this mode, the activities assume some additional, intermediate states. Note that the associated state transitions are neither recorded nor distributed as state change events. WfMOPEN supports the debug mode; it has been designed to run the workflow in a way that resembles the normal execution as closely as possible. The debug mode is not a simulation. It can best be compared to running a program in a debugger, i.e. the workflow is really executed. There are, however, predefined break points and the invocation of tools may be simulated instead of actually performed. Breakpoints are applied to monitoring effects on state changes, exceptions and deadlines.

F3 Execution, workflow engine

F3.1	Processes: Deployment, initialization, repositories	The WfMCore workflow engine component is capable of managing an unlimited number of process definitions at a time. When a new process definition file is imported, all currently managed descriptions with the same package id as the imported package (regardless of the process' name) are removed from the process definition directory before adding any new process definition. Replacing a process definition with another version (or removing a process definition) does not affect running processes. Once created, a process remains inseparably associated with the process definition that was effective when the process was created.
F3.2	Management of distributed workflows	Not provided
F3.3	Application Server Compatibility	As suggested by the engine documentation, WfmOpen runs on JBoss. It is the only tested environment.
F3.4	File formats	For the workflow execution are necessary only XPDL files.
F3.5	Transaction mechanisms	Transaction mechanisms are not directly supported. However, they can be built upon the XPDL subflows, which clearly define the interface between interactive processes. In all cases, the workflow operator has to implement mechanism to treat XPDL exceptions (supported by WfMOpen).
F3.6	Supported communication protocol	WfMOPEN supports SOAP, ASAP (Asynchronous SOAP), Corba, Wf-XML. Due to the exposing of API interfaces, other protocols can be implemented if necessary.
F3.7	Users' roles	User roles can be defined using an engine API called WfMCoreAdmin.

F4 Workflow client applications

F4.1	User interface technologies: client application - type, proprietary(heavy weight user clients) or general purpose (light weight browsers)	The workflow engine operates with client applications using SOAP communication or java API invocations provided by engine tools.

F5 Integration with other workflow engines; Supported standards

F5.1	Possibilities for collaboration with other workflow engines; standards	No explicit mechanism is present to collaborate with other workflow engines.

F6 Administration and monitoring

F6.1	Life cycle management	
F6.2	Supervising capabilities Process status	The engine offers an administration and monitoring tool which serves all typical functionalities (you can see Interface 5 section for more information).
F6.3	Flexibility to workflow participants	The engine allows participants administration. It can be done using Administration tool or Java API.

Functional Auxiliary Functions

FA1 Statistics
Supported by Administration tool

FA2 HELP Functionalities
Help information are present.

II Integral Characteristics

	WfMOpen				
Characteristics ISO/IEC 9126	**weak**	**good**	**strong**	**can't assess**	**score result**
Functionality	20%	20%	60%		**4,8**
Reliability		25%	75%		**5,5**
Usability	25%	37,5%	37,5%		**4,25**
Efficiency	67%	33%			**2,66**
Maintainability		60%	40%		**4,8**
Portability	20%	60%	20%		**4**

III Pie - Charts

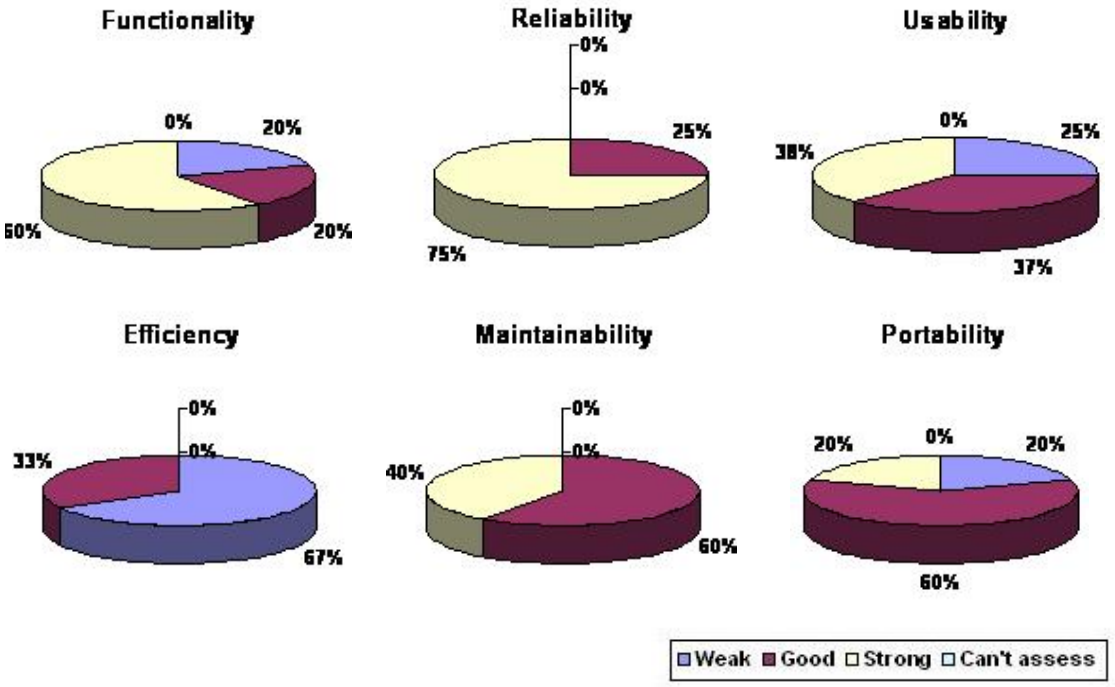


Figure 6.18 Pie-charts for WfMOpen's characteristics

YAWL

I. Functionality

Principal Functions

F1 Modelling, process definition

Fct no	Function name	Function implementation
F1.1	Process chart: definition, process thread	Each process definition consists of *tasks* (whether composite or atomic) and *conditions* which can be interpreted as places. Each process definition has one unique *input condition* and one unique *output condition* (see Figure 1, G1.3 and G.15 for details). In contrast to Petri nets, it is possible to connect 'transition-like objects' like composite and atomic tasks directly to each other without using a 'place-like object' (i.e., conditions) in-between. For the semantics this construct can be interpreted as a hidden condition, i.e., an implicit condition is added for every direct connection.
F1.2	Process: description, major, minor	Each task (either composite or atomic) can have multiple instances. Each task can have AND/XOR-splits/joins and OR-splits and OR-joins corresponding

Fct no	Function name	Function implementation
		respectively to 'Multi choice' pattern and 'Synchronizing merge' pattern. Finally, YAWL provides a notation for removing tokens from a specified region. The enabling of the task that will perform the cancellation may or may not depend on the tokens within the region to be "canceled". In any case, the moment this task executes, all tokens in this region are removed. This notation allows for various cancellation patterns.
F1.3	Event: initial, result, triggering process thread, event result correlation	The YAWL Engine performs the following steps: 1. loads a YAWL workflow specification 2. launches a YAWL specification 3. checks out available work items 4. submits/creates/suspend a checked out workitem The input condition is the entry point of each process. A process can be triggered by a client (user or web service) or by another running task (either via static binding or dynamic binding - worklet). The data is exchanged between tasks using data elements (variables) defined in both tasks. In order to exchange data between two tasks, they have to define variables having the same name and the same data type, declared as input/input&output type on one task and output/input&output type on another task.
F1.4	Flowline	The process flow is determined at run-time. The graphical modelling tool generates the static structure of the interactions between tasks. It is possible to modify dynamically the flow line at runtime, based on some selection condition evaluation by using worklets.
F1.5	Definition of organizational units	The YAWL engine deals with control-flow and data but not explicitly with users. In addition, the engine abstracts from differences between users, applications, organizations, etc. To achieve this, the YAWL system leverages principles from service-oriented architectures: external entities either offer services or require services. In a traditional workflow management system, the engine takes care of the 'What', 'When', 'How', and 'By whom'. In YAWL, the engine takes care of the 'What' and 'When' while the YAWL services take care of the 'How' and 'By whom'. By separating these concerns, it is possible to implement a highly efficient engine while allowing for customized functionality. It should be noted that the architecture of YAWL is similar to the architecture envisioned in the context of web service composition' languages like BPEL4WS, WSCI, BPML, etc. However, these languages typically only

Fct no	Function name	Function implementation
		consider control routing and data transformations, while YAWL is not limited to these aspects and addresses issues such as work distribution and management.
F1.6	Junctions	The YAWL supports AND/OR/XOR splits and joins of tasks.

F2 Simulation, debug

F2.1	Objects: definition, simulation arrival profile, access data, simulation process	The human user can interact with the YAWL Engine using a Web interface. The user can simulate the process run, can input values for data elements defined in the process (HTML forms are automatically generated for collecting the user input using Chiba XForms Processor).
F2.2	Process: initialization, service time profile, necessary resources , available resources, child diagram	See above
F2.3	Event: description, simulation event (generation, definition of object, simulation arrival profile)	See above
F2.4	Junction: description (and, or, x or), symbol, simulation outflow	See above
F2.5	Links: simulation probability, type	See above
F2.6	General conditions: simulation validation, run simulation	See above

F3 Execution, workflow engine

F3.1	Processes: Deployment, initialization, repositories	It is not possible to deploy the YAWL specifications to a YAWL Engine directly from the YAWL Editor. The YAWL specifications should be loaded using the YAWL Web interface (or programmatically).
F3.2	Management of distributed workflows	Supported via WS interface; it is possible to have any number of YAWL Engine running and communication between them is achieved by exchanging HTML/XML messages. Note that the same engine does not support the allocation of tasks to separate resources. i.e., the traditional resource perspective of workflow is not yet supported.
F3.3	Application Server Compatibility	Apache Tomcat 5.0
F3.4	File formats	YAWL XML Format
F3.5	Transaction mechanisms	Not applicable
F3.6	Supported communication	HTML/WSDL

	protocol	
F3.7	Users' roles	It is possible to create an organization model, with users and other type of resources, using the Web interface.

F4 Workflow client applications

F4.1	User interface technologies: client application - type, proprietary(heavy weight user clients) or general purpose(light weight browsers)	Any Web browser can be used to administrate and to operate the YAWL Engine.

F5 Integration with other workflow engines; Supported standards

F5.1	Possibilities for collaboration with other workflow engines; standards	Not supported

F6 Administration and monitoring

F6.1	Life cycle management	Not supported
F6.2	Supervising capabilities Process status	What kind of services, monitoring: active processes, alarm queue, Failures
F6.3	Flexibility to workflow participants	YAWL Organizational model supports the management of users, roles and other resources.

Functional Auxiliary Functions

FA1 Statictics
Not supported

FA2 HELP Functionalities

Detailed Installation Manual and User Guide are available for YAWL Engine and YAWL Worklet. Several documents and white papers are available for download from the YAWL website. The white papers are organized in two levels: an academic (more theoretical) level and a system (practical oriented) level.
No help available for YAWL Editor

II Integral Characteristics

	YAWL Engine				
Characteristics ISO/IEC 9126	**weak**	**good**	**strong**	**can't assess**	**score result**
Functionality	40%		40%	20%	**3,2**
Reliability	100%				**2**
Usability	37,5%	37,5%	25%		**3,75**
Efficiency	50%	50%			**3**
Maintainability	40%		60%		**4,4**
Portability	20%		40%	40%	**2,8**

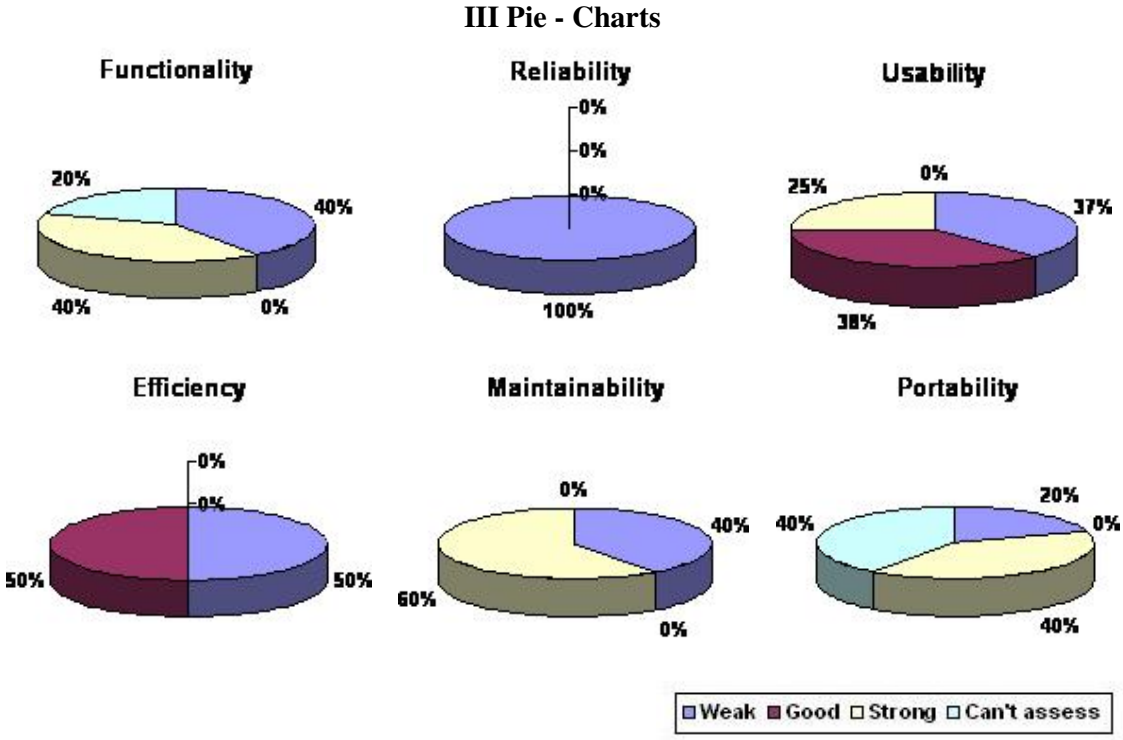


Figure 6.19 Pie-charts for YAWL's characteristics

6.4 Comparisons

This section presents comparative diagrams of the assessed workflow products.

Column diagrams

This paragraph performs a decomposition of the software products, according to their general workflow functionality. Four general categories of the products have been defined:

- Modelling Tools
- Workflow Editors;
- Workflow Engines;
- Combined Products: Editor + Engine

The workflow software products have been evaluated and compared between each other for the appropriate functional category. Modelling tools conform to other standards (BPMN, UML) than workflow orchestration tools. They are included in additional category. Thus, three different comparisons have been applied, relating to the functional category of the product. Appropriate diagrams are designed. These diagrams demonstrate the advantages of a chosen product in comparison with its corresponding functional peers.

Following this decomposition scheme and giving importance to the functional quality of the products, the assessment of the products are performed for four groups of products presented in the table below.

Table 6.1. General workflow functionality decomposition of the products

Categories	Workflow Software Products
Modelling Tools	Borland Together Designer Process Modeler UML Modelling Tools
Workflow Editors	ActiveWebflow Designer Cape Clear con-cern enterpriseXtention Enhydra JaWE JBoss jBPM ObjectWeb BONITA OpenWFE Oracle BPEL Process Manager SAP NetWeaver Exchange Infrastructure YAWL
Workflow Engines	ActiveBPEL Engine ActiveWebflow Standard Biztalk Server Cape Clear con-cern Enhydra Shark enterpriseXtention Freefluo JBoss jBPM ObjectWeb BONITA OpenWFE Oracle BPEL Process Manager SAP NetWeaver Exchange Infrastructure WfMOpen YAWL
Combined Products Editor + Engine	ActiveWebflow Designer + ActiveWebflow Standard Cape Clear con-cern Enhydra JaWE + Enhydra Shark enterpriseXtention JBoss jBPM ObjectWeb BONITA OpenWFE Oracle BPEL Process Manager SAP NetWeaver Exchange Infrastructure YAWL

The comparative assessment of the products, belonging to the category “Workflow editors” and addressing their functional rating is given to the bar chart diagram.

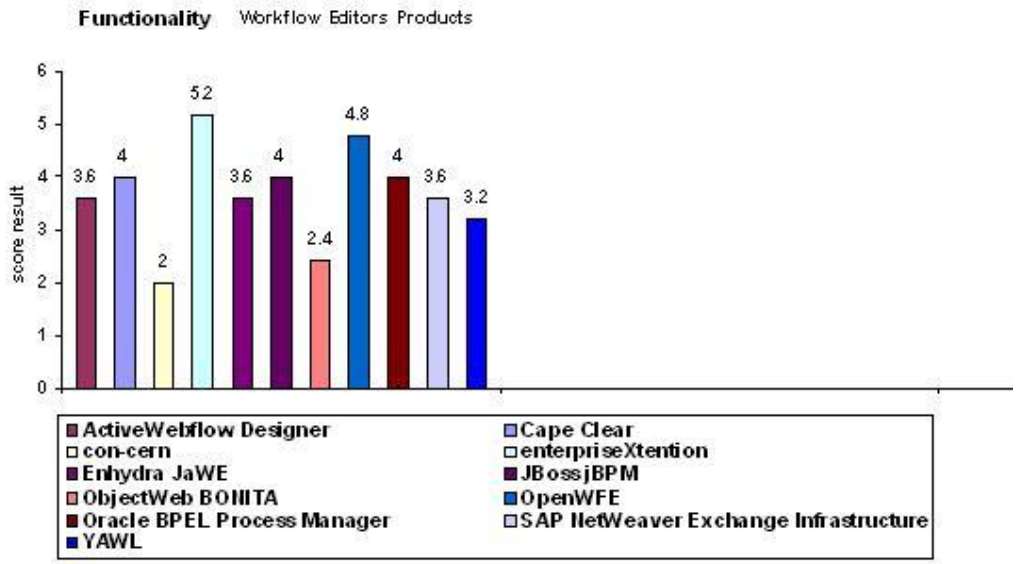


Figure 6.20 Workflow Editors comparative chart

According to this diagram, potential candidates and winner for application in the VISP developments are the products with score around 4 and supporting workflow standards, recommened by D2.1 (BPEL, XPDL). These products are Active Webflow Designer, Cape Clear, Enhydra JaWE, and Oracle BPEL Process Manager. Their personal rates, concerning the functional and usability software quality are presented on the pie chart diagrams.

The functional comparison between the products, belonging to the category Workflow engines" is given below.

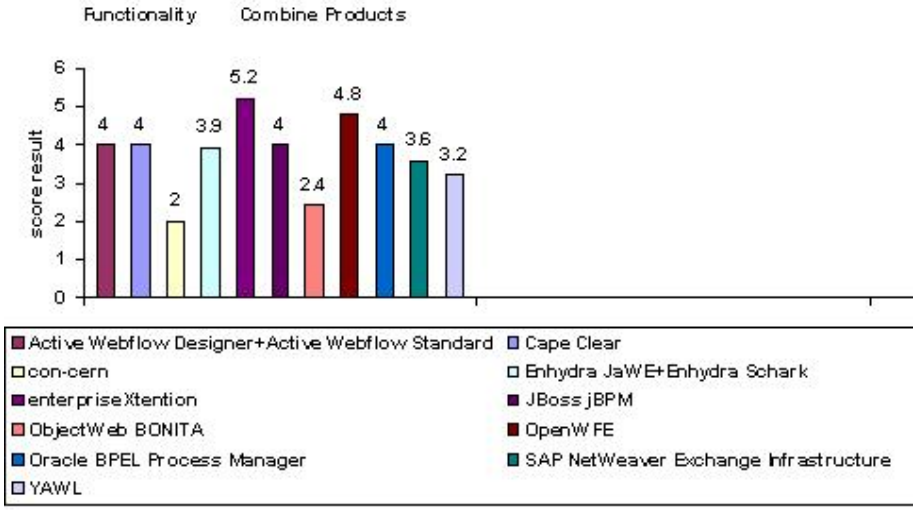


Figure 6.21 Workflow Engines comparative chart

This diagram points the potential candidates and "winner" for application in the VISP development according to the products functional rate, which is above the score 4 and

according to recommended standards. These products are: Active BPEL Engine, Active Webflow Standard, Cape Clear, Enhydra Shark, Oracle BPEL Process Manager.

The last diagram summarizes the rates of the products from the category of mutual support of editing and execution functionality.

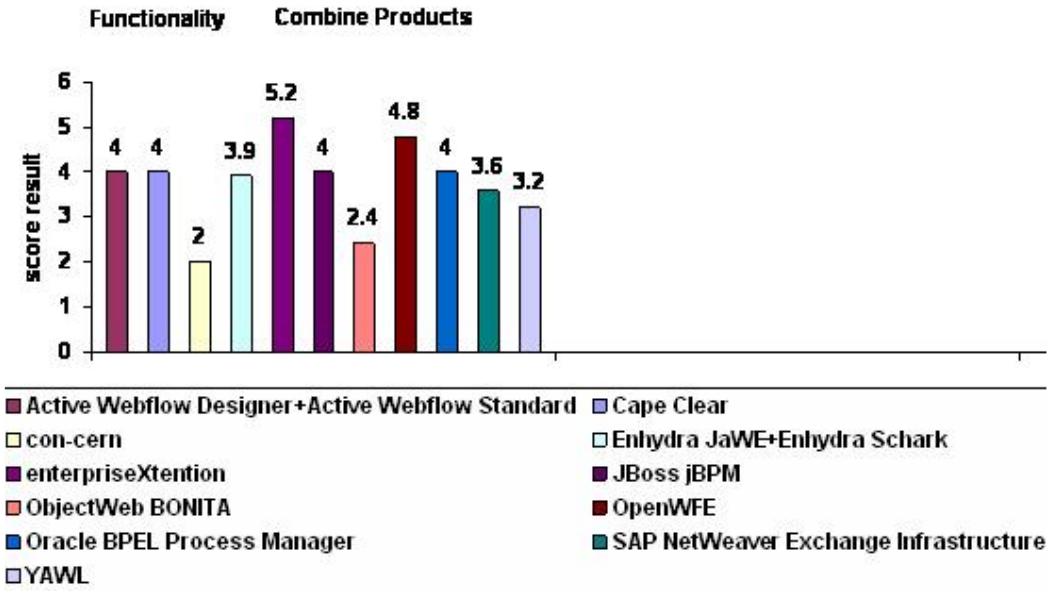


Figure 6.22 Workflow Editors + Engines comparative chart

The prospective products have scored more than 4 and they are: ActiveWebflow Standard + ActiveWebflow Designer, Cape Clear, Enhydra Jawe+Shark, Oracle BPEL Process Manager. An evaluation of the products per category Editor, Engine, Editor+Engine is performed for the software quality Usability. The corresponding block diagrams are:

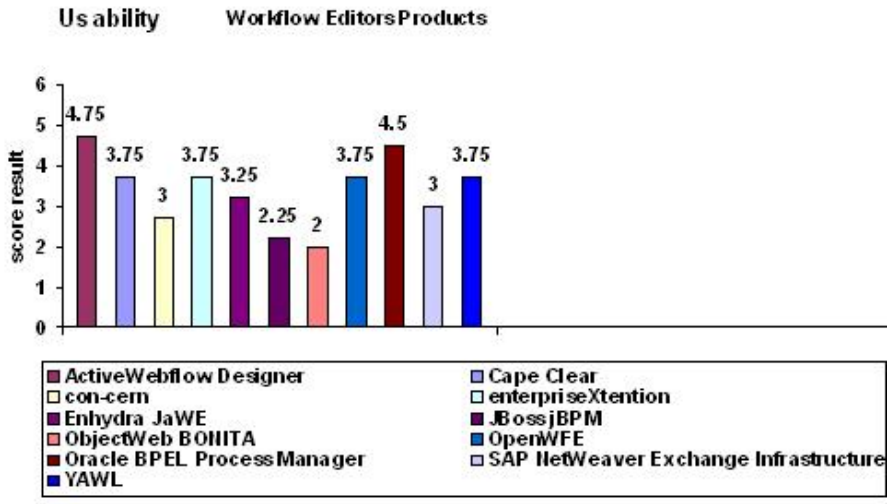


Figure 6.23 Workflow Editors Usability comparative chart

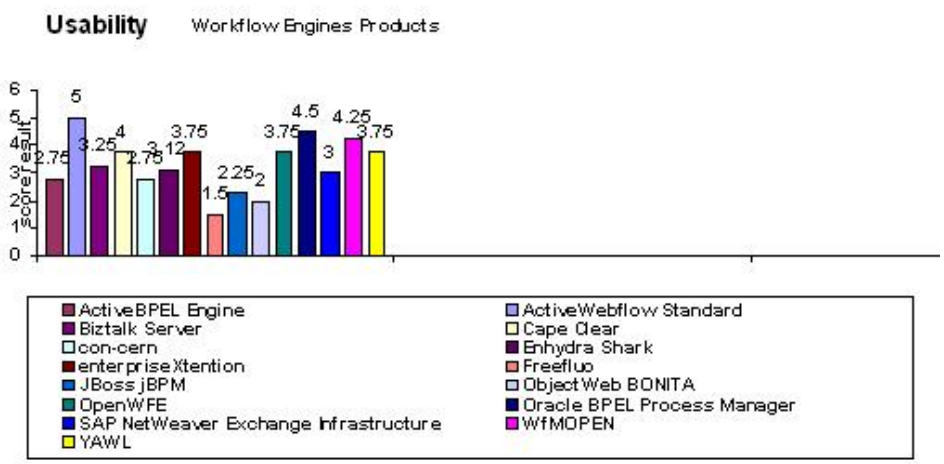


Figure 6.24 Workflow Engines Usability comparative chart

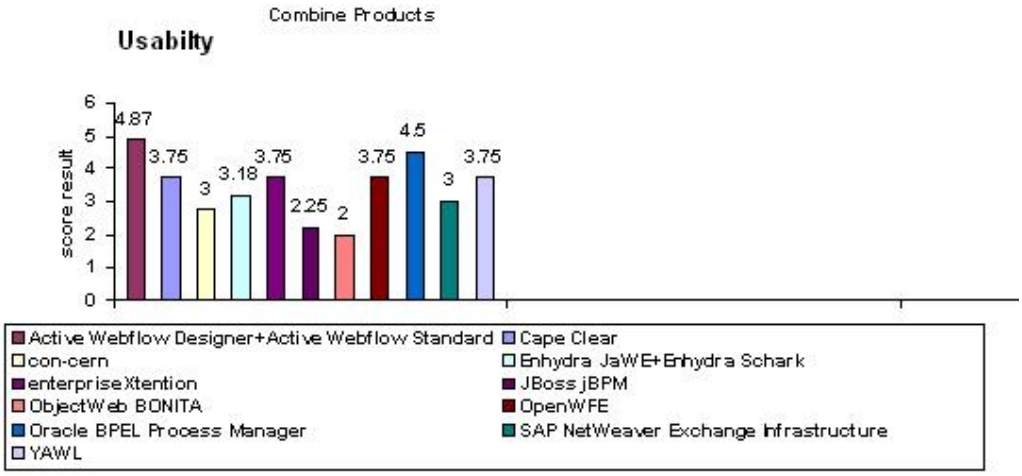


Figure 6.25 Workflow Editors + Engines Usability comparative chart

For integral evaluation of the quality of the tested product, a common comparison has been performed for the other main software quality categories: reliability, efficiency, maintainability, portability. The comparative assessments are given below.

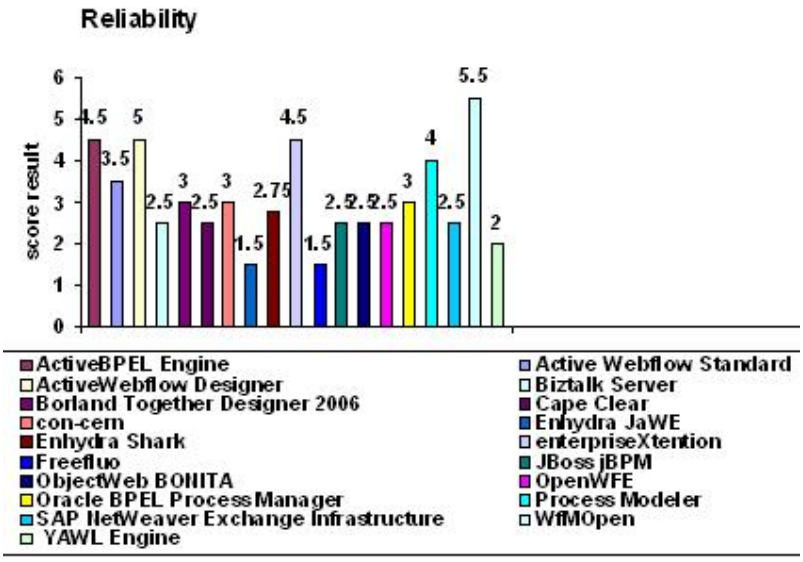


Figure 6.26 Reliability comparative chart

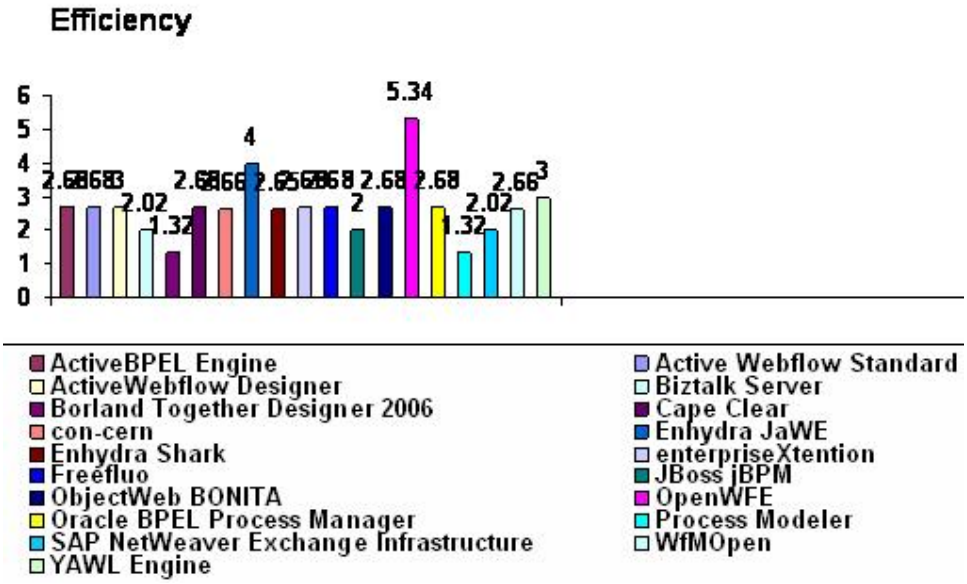


Figure 6.27 Efficiency comparative chart

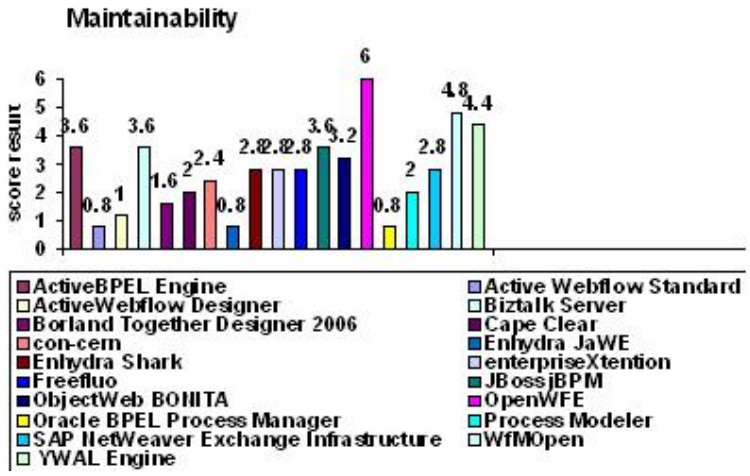


Figure 6.28 Maintainability comparative chart

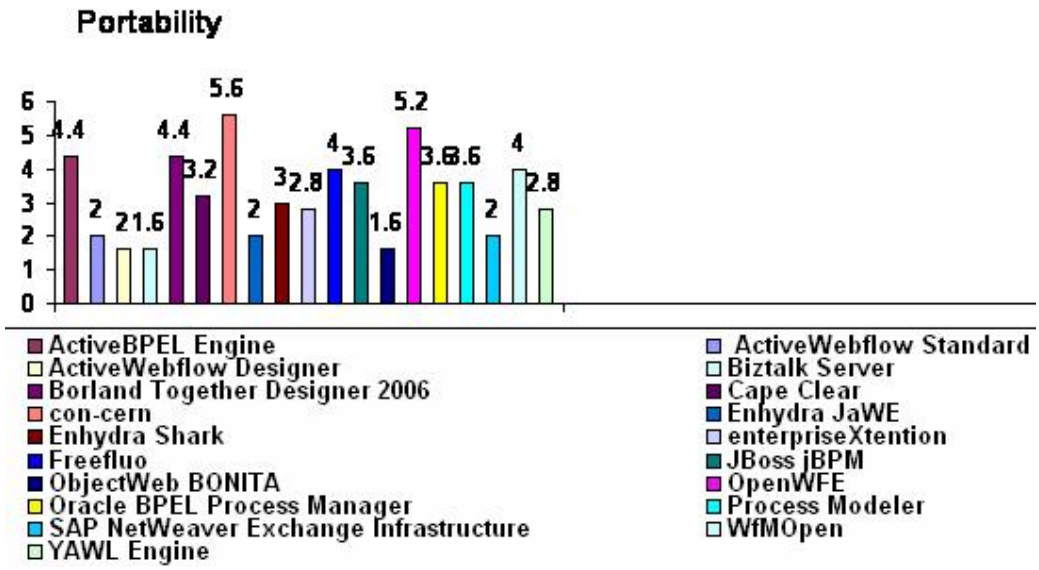


Figure 6.29 Portability comparative chart

The column diagrams give integral view of the quality characteristics of the tested products. Despite that, they introduce quantitative approach for the comparison and assessment, the problem of the choice is still vector one, which does not allow finding narrow set of prospective candidates for practical use in the VISP project developments.

An approach to find unique solutions of the problem of vector choice of product is to weight the general software quality characteristics.

Radar diagrams

This study tries to find a narrow set of prospective workflow software products, by using the test results of the products. The average evaluation results per product is summarize in a table below.

Table 6.2. Average evaluation results per products

	Active BPEL Engine	ActiveWebflow™ Standard	ActiveWebflow™ Professional Designer	BizTalk Server	Borland Together	Cape Clear 6.5	con:cern	Enhydra JaWE	Enhydra Shark	enterprise Xtention
Characteristics ISO/IEC 9126	**score result**	**score result**	**score result**	**score result**	**score result**	**score result**	**score result**	**score result**	**score result**	**score result**
Functionality	**4,8**	**4,4**	**3,6**	**2,8**	**4**	**4**	**2**	**3,6**	**4,2**	**5,2**
Reliability	**4,5**	**3,5**	**4,5**	**2,5**	**3**	**2,5**	**3**	**1,5**	**2,75**	**4,5**
Usability	**2,75**	**5**	**4,75**	**3,25**	**4,75**	**3,75**	**2,75**	**3,25**	**3,125**	**3,75**
Efficiency	**2,68**	**2,68**	**2,68**	**2,02**	**1,32**	**2,68**	**2,66**	**4**	**2,65**	**2,68**
Maintainability	**3,6**	**0,8**	**1,2**	**3,6**	**1,6**	**2**	**2,4**	**0,8**	**2,8**	**2,8**
Portability	**4,4**	**2**	**1,6**	**1,6**	**4,4**	**3,2**	**5,6**	**2**	**3**	**2,8**

	FreeFluo	jBPM	Bonita	OpenWFE	Oracle BPEL Process Manager	ProcessModeler	SAP_NetWeaver	WfMOpen	YAWL	Products
Characteristics ISO/IEC 9126	**score result**	**score result**	**score result**	**score result**	**score result**	**score result**	**score result**	**score result**	**score result**	**Average value**
Functionality	**3,2**	**4**	**2,4**	**4,8**	**4**	**3,6**	**3,6**	**4,8**	**3,2**	**3,82**
Reliability	**1,5**	**2,5**	**2,5**	**2,5**	**3**	**4**	**2,5**	**5,5**	**2**	**3,05**
Usability	**1,5**	**2,25**	**2**	**3,75**	**4,5**	**4**	**3**	**4,25**	**3,75**	**3,46**
Efficiency	**2,68**	**2**	**2,68**	**5,34**	**2,68**	**1,32**	**2,02**	**2,66**	**3**	**2,65**
Maintainability	**2,8**	**3,6**	**3,2**	**6**	**0,8**	**2**	**2,8**	**4,8**	**4,4**	**2,74**
Portability	**4**	**3,6**	**1,6**	**5,2**	**3,6**	**3,6**	**2**	**4**	**2,8**	**3,2**

The following subtables present these scores again, but for each category:

a) Modelling tools,

b) Workflow editors,

c) Workflow engines and

d) Combined products

Table 6.3. Average evaluation results – Modelling Tools

	Borland Together	ProcessModeler	Products
Characteristics ISO/IEC 9126	**score result**	**score result**	**Average value**
Functionality	**4**	**3,6**	**3,82**
Reliability	**3**	**4**	**3,05**
Usability	**4,75**	**4**	**3,46**
Efficiency	**1,32**	**1,32**	**2,65**
Maintainability	**1,6**	**2**	**2,74**
Portability	**4,4**	**3,6**	**3,2**

Table 6.4. Average evaluation – Workflow editors

	ActiveWebflow™ Professional Designer	Cape Clear 6.5	con:cern	Enhydra JaWE	enterprise Xtention	jBPM
Characteristics ISO/IEC 9126	score result	score result	score result	score result	score result	score result
Functionality	3,6	4	2	3,6	5,2	4
Reliability	4,5	2,5	3	1,5	4,5	2,5
Usability	4,75	3,75	2,75	3,25	3,75	2,25
Efficiency	2,68	2,68	2,66	4	2,68	2
Maintainability	1,2	2	2,4	0,8	2,8	3,6
Portability	1,6	3,2	5,6	2	2,8	3,6

	Bonita	OpenWFE	Oracle BPEL Process Manager	SAP_Net Weaver	YAWL	Products
Characteristics ISO/IEC 9126	score result	score result	score result	score result	score result	Average value
Functionality	2,4	4,8	4	3,6	3,2	3,82
Reliability	2,5	2,5	3	2,5	2	3,05
Usability	2	3,75	4,5	3	3,75	3,46
Efficiency	2,68	5,34	2,68	2,02	3	2,65
Maintainability	3,2	6	0,8	2,8	4,4	2,74
Portability	1,6	5,2	3,6	2	2,8	3,2

Table 6.5. Average evaluation results – Workflow Engines

	Active BPEL Engine	ActiveWebflow™ Standard	BizTalk Server	Cape Clear 6.5	con:cern	Enhydra Shark	enterprise Xtention	FreeFluo
Characteristics ISO/IEC 9126	score result	score result	score result	score result	score result	score result	score result	score result
Functionality	4,8	4,4	2,8	4	2	4,2	5,2	3,2
Reliability	4,5	3,5	2,5	2,5	3	2,75	4,5	1,5
Usability	2,75	5	3,25	3,75	2,75	3,125	3,75	1,5
Efficiency	2,68	2,68	2,02	2,68	2,66	2,65	2,68	2,68
Maintainability	3,6	0,8	3,6	2	2,4	2,8	2,8	2,8
Portability	4,4	2	1,6	3,2	5,6	3	2,8	4

	jBPM	Bonita	OpenWFE	Oracle BPEL Process Manager	SAP_Net Weaver	WfMOpen	YAWL	Products
Characteristics ISO/IEC 9126	score result	score result	score result	score result	score result	score result	score result	Average value
Functionality	4	2,4	4,8	4	3,6	4,8	3,2	3,82
Reliability	2,5	2,5	2,5	3	2,5	5,5	2	3,05
Usability	2,25	2	3,75	4,5	3	4,25	3,75	3,46
Efficiency	2	2,68	5,34	2,68	2,02	2,66	3	2,65
Maintainability	3,6	3,2	6	0,8	2,8	4,8	4,4	2,74
Portability	3,6	1,6	5,2	3,6	2	4	2,8	3,2

Table 6.6. Average evaluation – Combined Products (Editor+Engines)

	ActiveWebflow ™ Standard + ActiveWebflow ™ Professional Designer	Cape Clear 6.5	con:cern	Enhydra JaWE + Enhydra Shark	enterpriseXtention
Characteristics ISO/IEC 9126	**score result**	**score result**	**score result**	**score result**	**score result**
Functionality	**4**	**4**	**2**	**3.9**	**5,2**
Reliability	**4**	**2,5**	**3**	**2.13**	**4,5**
Usability	**4.88**	**3,75**	**2,75**	**3.19**	**3,75**
Efficiency	**2,68**	**2,68**	**2,66**	**3.33**	**2,68**
Maintainability	**1**	**2**	**2,4**	**1.9**	**2,8**
Portability	**1.8**	**3,2**	**5,6**	**2.5**	**2,8**

	jBPM	Bonita	OpenWFE	Oracle BPEL Process Manager	SAP_Net Weaver	YAWL	Products
Characteristics ISO/IEC 9126	**score result**	**score result**	**score result**	**score result**	**score result**	**score result**	**Average value**
Functionality	**4**	**2,4**	**4,8**	**4**	**3,6**	**3,2**	**3,82**
Reliability	**2,5**	**2,5**	**2,5**	**3**	**2,5**	**2**	**3,05**
Usability	**2,25**	**2**	**3,75**	**4,5**	**3**	**3,75**	**3,46**
Efficiency	**2**	**2,68**	**5,34**	**2,68**	**2,02**	**3**	**2,65**
Maintainability	**3,6**	**3,2**	**6**	**0,8**	**2,8**	**4,4**	**2,74**
Portability	**3,6**	**1,6**	**5,2**	**3,6**	**2**	**2,8**	**3,2**

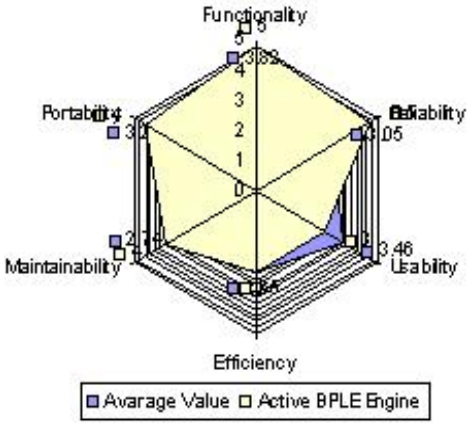
Having the results from the evaluation, an average score per each software quality for all products can be found. Thus, the average mark for all products for the category Functionality is calculated as:
Functionality (average) = [Functionality (product1)+ … + Functionality(productN)]/N, where N is the number of the tested product.

By the same way, appropriate average values have been calculated per each software category: reliability, usability, efficiency, maintainability, portability.

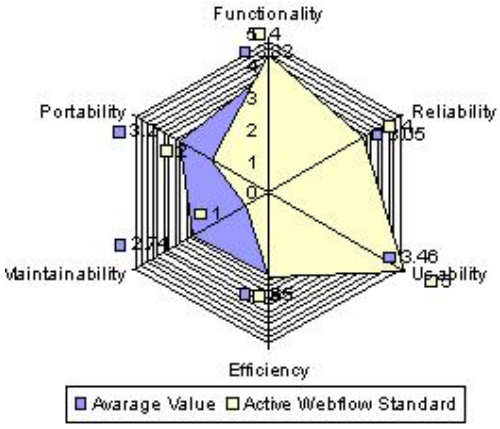
Then every tested product has been compared with the "average" results by means to identify the difference between the product quality and the "average" products scores. The results have been presented in a form of radar diagrams. The axis of the diagrams consist the main software quality characteristics. The area, covered by the product from the radar diagram is a measure for the potential and the performance of the product. The particular product is compared with the area, covered by the average values for the tested products. Thus, the prospective product can be identified by the overlapping regions of the radar diagrams. The close overlap defines the better software product.

The particular results for the product assessment according to the average values of the testing are given per product in a radar diagram form.

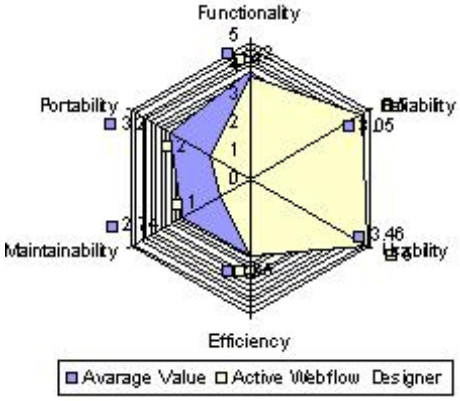
ActiveBPEL Engine



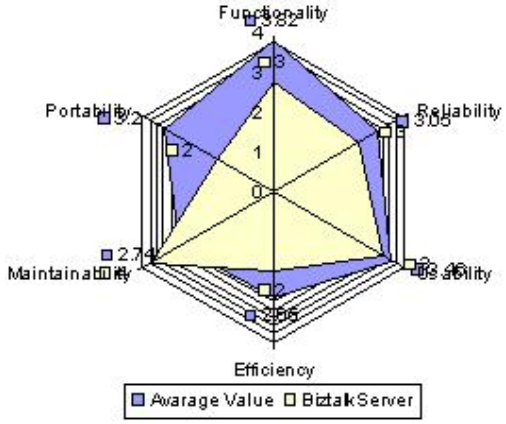
ActiveWebflow Standard



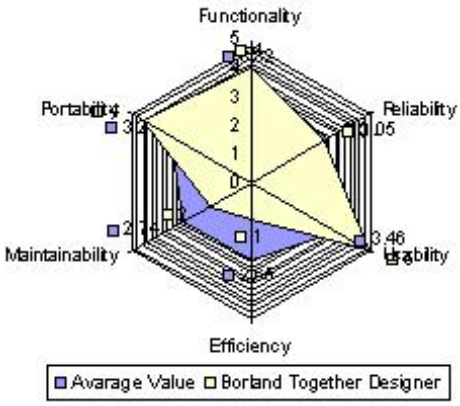
ActiveWebflow Designer



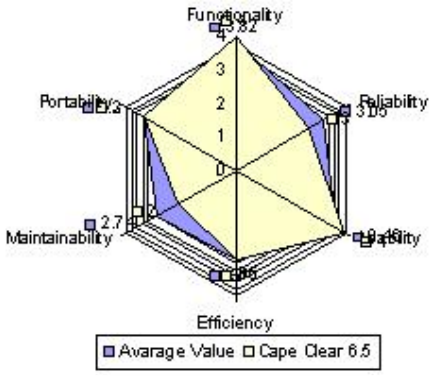
Biztalk Server



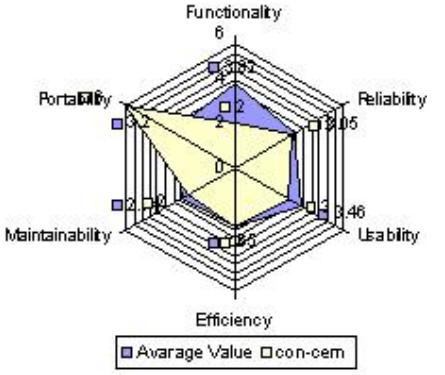
Borland Together Designer 2006



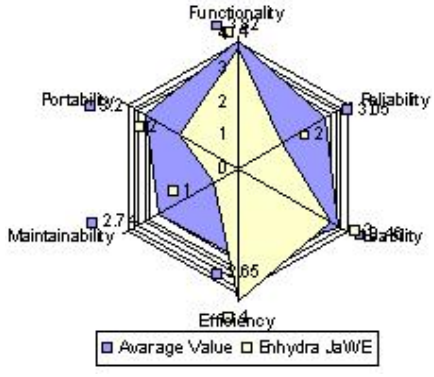
Cape Clear



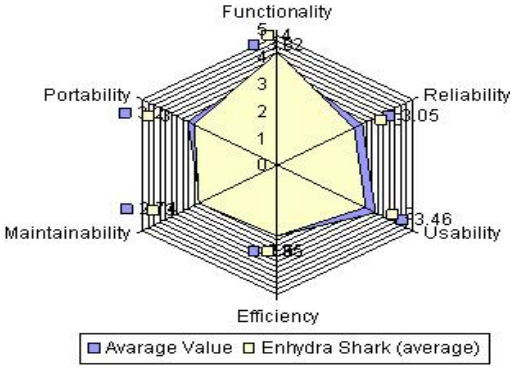
Con-cern



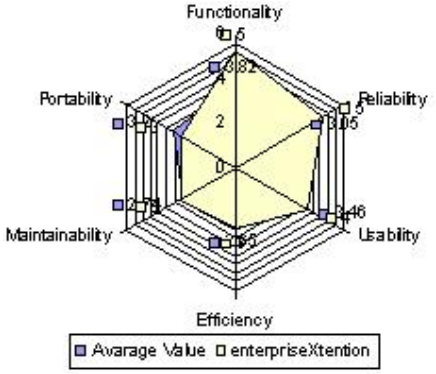
Enhydra JaWE



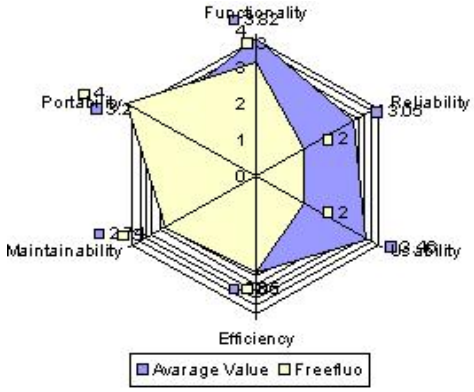
Enhydra Shark



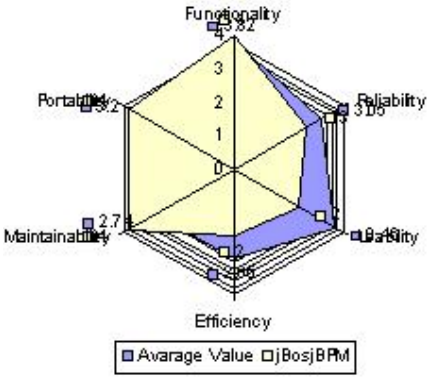
EnterpriseXtention



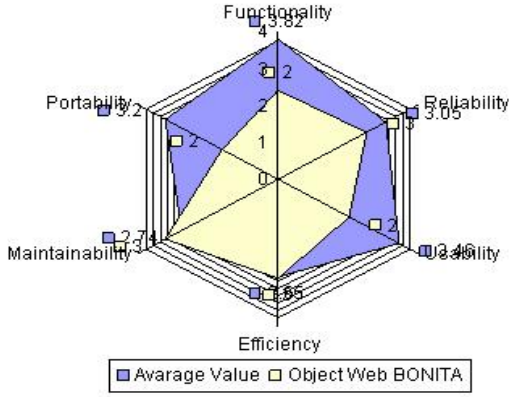
Freefluo



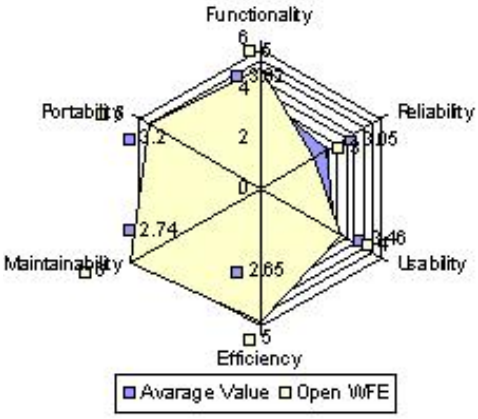
JBoss jBPM



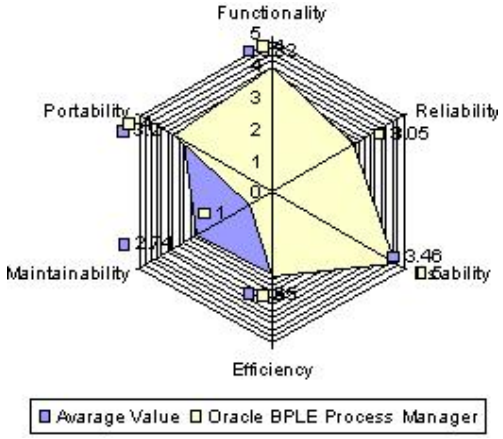
ObjectWebBonita



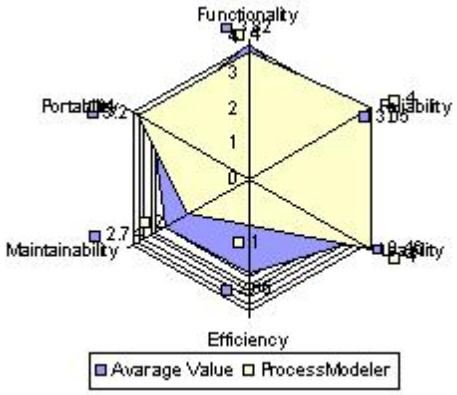
OpenWFE



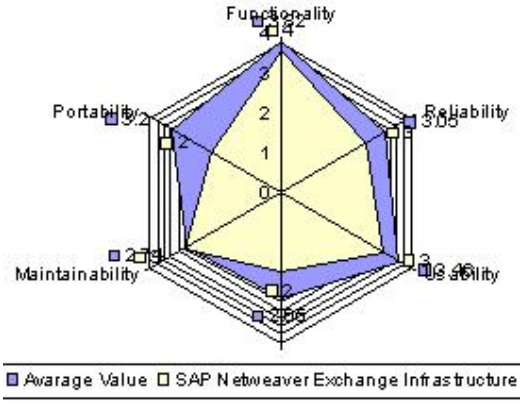
Oracle BPEL Process Manager



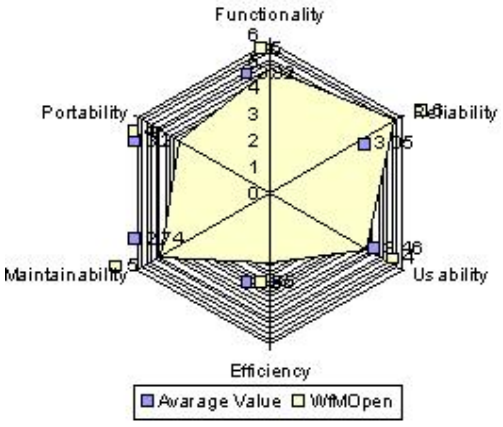
Process Modeler



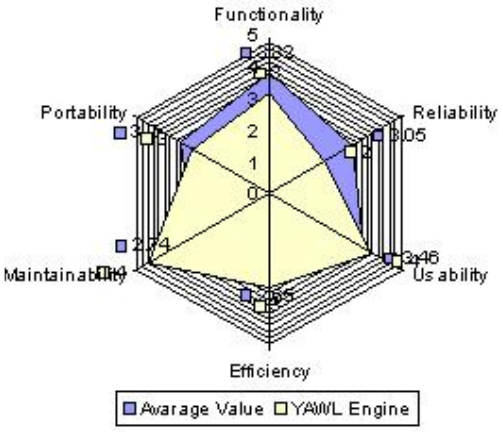
SAP NetWeaver Exchange Infrastructure



WfMOpen



YAWL



Following the drawings of the radar diagrams, the prospective candidates are the software products: Active BPEL Engine, Active Webflow Designer, Cape Clear, Enhydra Shark, FreeFlue, and Yawl.

6.5 Conclusions

The evaluations and comparisons between the software products, which are targeted for implementation in the VISP project allows narrowing the choice to a restricted set of products. An advantage of the product is this one, related to it functionality to cover both editing and execution operations for the workflow design and exploitation. The prospective set of products is defined as:

- Modelling tools: Borland Together Designer, Process Modeler
- Editing tools: Borland Together Designer, Cape Clear, JBoss jBPM and Oracle BPEL Process Manager

- Execution tools: Active BPEL Engine, Active Webflow Standard, Cape Clear, Enhidra Shark, Oracle BPEL Process Manager;
- Both modelling and execution tools: ActiveWebflow Standard + ActiveWebflow Designer, Cape Clear, Enhydra Jawe+Shark, Oracle BPEL Process Manager.

The average comparison between the products identifies the prospective tools as Active BPEL Engine, Active Webflow Designer, Cape Clear, Enhydra Shark and FreeFlue, Yawl. The supported workflow standardization approach per products is presented in the tables below.

Table 6.7. Modelling Tools

Software Products	Covered Technologies	Comments
Borland Together Designer	BPMN, UML	Borland Together supports the *specification* of UML and BPMN. It supports *import* of WSDL and associated XML and it *exports* BPEL.
Process Modeler	BPMN	Process Modeler imports WSDL and exports BPEL and XPDL

Table 6.8. Workflow Editors

Workflow Software Products	Covered Technologies	Comments
ActiveWebflow Designer	XML, BPEL, WSDL	AWD supports BPEL standard. A process can be created by either using existing WSDL files or building an abstract BPEL and then create the WSDL files for it. Data manipulation is accomplished through XML variables. The links connecting activities use XPath expression.
Cape Clear	XML, BPEL, WSDL	
Enhydra JaWE	XPDL, Wf-XML, LDAP	JaWE supports XPDL as its native file format, Wf-XML standard and LDAP connections. A process can be imported from external package and participants can be imported from LDAP server.
Oracle BPEL Process Manager	XML,BPEL, WSDL	Oracle as an editor supports the *specification* of BPEL utilizing XPath and WSDL incl. associated XML if necessary. BPEL, WSDL, and XML can be *imported* and *exported*. It utilizes UDDI and WS_Inspection for the discovery of WS. As an engine it supports BPEL that has been compiled to Java code. It utilizes SOAP and WS_Adressing and proprietary Java mechanisms.

Table 6.9. Workflow Engines

Workflow Software Products	Covered Technologies	Comments
ActiveBPEL Engine	XML, BPEL, WSDL, SOAP, XPath, XQuery	The ActiveBPEL engine can import either 1.X or 2.0 Process Deployment Descriptor (PDD), Partner Definitions (PDEF), WSDL BPEL and BPRD deployment files. The workflow engine can be invoked using distribution java API or by SOAP calls (synchronous and asynchronous communication are supported). Import/export data towards other products like databases, file systems, repositories (UDDI, LDAP); monitoring of transactions.
ActiveBPEL Standard	XML, BPEL, WSDL, SOAP, XPath, XQuery	AWS uses WSDL and BPEL for specifying services and processes, XML data files, SOAP for communication. Service location is specified using WS-Addressing. XPath expression is within BPEL files.
Cape Clear	XML, BPEL, WSDL, SOAP	
Enhydra Shark	XPDL, Wf-XML, SOAP, ASAP	Supports SOAP, ASAP (Asynchronous SOAP), Corba; Wf-XML Based on WfMC specifications using XPDL as its native workflow process definition format, WfMC "ToolAgents" API for serverside execution, Wf-XML standard, XPDL, XML file formats.
Oracle BPEL Process Manager	XML,BPEL, WSDL, SOAP, UDDI, XPath	Oracle as an editor supports the *specification* of BPEL utilizing XPath and WSDL incl. associated XML if necessary. BPEL, WSDL, and XML can be *imported* and *exported*. It utilizes UDDI and WS_Inspection for the discovery of WS. As an engine it supports BPEL that has been compiled to Java code. It utilizes SOAP and WS_Adressing and proprietary Java mechanisms.

Table 6.10. Combined Products: Editor + Engine

Workflow Software Products	Covered Technologies
ActiveWebflow Designer + ActiveWebflow Standard	XML, BPEL, WSDL, SOAP, XPath, XQuery
Cape Clear	XML, BPEL, WSDL, SOAP
Enhydra JaWE + Enhydra Shark	XPDL, Wf-XML, SOAP, ASAP
Oracle BPEL Process Manager	XML,BPEL, WSDL, SOAP, UDDI, XPath

These results remind that the unique choice of product could not be the right strategy for implementation. The benefit of the existence of different and wide available products has to be consumed by allocating rights for the choice for the different workflow partners in the workflow integration. However, the definition and exploitation of a common business distributed workflow system has to respect common standards for modelling and execution of the workflow tasks. Thus, the integration can be achieved, despite the different software products, which are used, but based on a common used standardization and methodological approach.

7. CONCLUSIONS

This deliverable contains the results of the analysis and comparison of different business workflow products, which are implemented for the stages of modelling, simulation and exploitation of workflow management systems. The diversity of these technologies and of the existing workflow specification activities makes difficult the right technological decision about the two key aspects that affect the entire life cycle of any eBusiness development: the choice of a choreography language and of an orchestration language. Respectively to find the prospective products, implementing such modelling and execution tasks is an ambitious and not easy task.

The deliverable D2.2 performs a state-of-the-art analysis with a comparison of workflow software products. A special emphasis is placed on the comparison of existing Open Source software supporting the different workflow technologies since the project intends to integrate existing software solutions, not to redevelop what already exists.

The developments in the deliverable begin with the identification of the importance of the automation for the design, implementation and exploitation of workflow management systems. An extensive survey is performed about the products, software suits and programs, which deal with different aspects of the workflow management. Classifications of these products have been done, concerning their internal software origin, their sets of functionalities and targeted domains of implementation. Because of the different functional nature of the products, the problem for the evaluation and comparison of these products has to be resolved. It has been analyzed the current experience addressing the comparison and evaluation of different software products and the methodological solutions applied for the assessment of relatively different software suits. The lack of direct quantitative evaluations of the products required the works in this deliverable to be directed towards getting personal experience of them. Thus, the deliverable work involve installation, configuration and trial test of a several software products, chosen among the set of candidate technologies.

The products have been tested and personal experience has been got about the product potential. An evaluation methodological scheme was designed, which minimizes the subjective influence of the experts for their personal evaluation findings. A common evaluation scheme, based on objective requirements towards the products, derived from the standard ISO/IEC 9126 for the quality of the software products was designed. Following the requirements of the standard, a template has been designed for a common framework of the evaluation of the software products. The results from the assessment have been summarized and classified. The products have been classified in four general functional groups:

- Modelling Tools
- Workflow Editors;
- Workflow Engines;
- Combined Products: Editor + Engine

For each group appropriate weighted evaluation scheme has been performed. The prospective products were identified as:

- **Modelling tools**: The modelling tools, which are prospective for the next VISP developments could be the products, based on BPMN and UML standardization methodology. Particular interest for the project developments are the tested products Borland Together Designer, Process Modeler, UML Tools.

- **Editing tools**: Borland Together Designer, Cape Clear, JBoss jBPM and Oracle BPEL Process Manager

- **Execution tools**: Active BPEL Engine, Active Webflow Standard, Cape Clear, Enhidra Shark, Oracle BPEL Process Manager

- **Both editing and execution tools**: ActiveWebflow Standard + ActiveWebflow Designer, Cape Clear, Enhydra Jawe+Shark, Oracle BPEL Process Manager.

These results remind that the unique choice of product could not be the right strategy for implementation. The benefit of the existence of different and wide available products has to be consumed by allocating rights for the choice for the different workflow partners in the workflow integration. However, the definition and exploitation of a common business distributed workflow system has to respect common standards for modelling and execution of the workflow tasks. Thus, the integration can be achieved, despite the different software products, which are used, but based on a common used standardization and methodological approach. Taking into account the supported standardization and technological support, prospective technologies for the VISP project developments appear to be XPDL and BPEL. Respectively the products, which support them can be used in particular project solutions.

REFERENCES

[2.1] http://wfmc.org/
[2.2] http://wfmc.org/standards/model.htm
[2.3] http://www.wfmc.org/standards/docs/TC-1025_10_xpdl_102502.pdf
[2.4] http://www.jcp.org/en/jsr/detail?id=207
[2.5] http://www.bpmi.org
[2.6] http://ebxml.org/specs/ebBPSS.pdf
[2.7] http://www.bpmi.org/
[2.8] http://www-128.ibm.com/developerworks/library/specification/ws-bpel/
[2.9] http://www-128.ibm.com/developerworks/webservices/library/ws-bpelj/
[2.10] http://www.omg.org/docs/formal/00-05-02.pdf
[2.11] http://www.uml.org/
[2.12] http://www.rosettanet.org/RosettaNet/Public/PublicHomePage
[2.13] http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=ubl
[3.1] http://en.wikipedia.org/wiki/Workflow
[3.2] http://en.wikipedia.org/wiki/Workflow
[3.3] http://www.manageability.org/blog/stuff/ workflow_in_java/view
[3.4] http://www.gripopprocessen.nl/index.php? id=35&no_cache=1
[3.5] http://java-source.net/open-source/workflow-engines
[3.6] http://e-docs.bea.com/wlintegration/v2_1/ interm/bpmhome.htm
[3.7] http://www.carnot-usa.com/
[3.8] http://www.dralasoft.com/products/workflow/
[3.9] http://www.filenet.com/
[3.10] http://www.fsw.fujitsu.com/products/InterstageSuite/ BPM/overview.html
[3.11] http://www-128.ibm.com/developerworks/ibm/library/i-holo/
[3.12] http://www.intalio.com/products/index.html
[3.13] http://www.lombardisoftware.com/
[3.14] http://oakgrovesystems.com/
[3.15] http://www.oracle.com/index.html
[3.16] http://www.qlinktech.com/
[3.17] http://www.sap.com/solutions/netweaver/index.epx
[3.18] http://www.savvion.com/products/
[3.19] http://www.seebeyond.com/software/ican.asp
[3.20] http://www.sonicsoftware.com/products/ sonic_orchestration_server/index.ssp
[3.21] http://www.staffware.com/
[3.22] http://www.ultimus.com/index
[3.23] http://www.versata.com/products/index.htm
[3.24] http://www.webmethods.com/meta/default/ folder/0000005452
[3.25] http://is.tm.tue.nl/research/patterns/
[3.26] http://info.lib.uh.edu/pr/v8/n3/smit8n3.html
[3.27] http://www.axia-consulting.co.uk/index.html
[3.28] http://www.library.cornell.edu/olinuris/ref/research/ webcrit.html
[3.29] http://www.cyberbee.com/content.pdf
[3.30] http://www.cyberbee.com/design.pdf
[3.31] http://www.lanner.com/,
[3.32] http://www.mega.com
[3.33] http://www.intalio.com
[3.34] http://www.fujitsu.com/global/services/software/interstage/products/bpm/
[3.35] http://www.axway.com

[3.36] http://www.active-flow.com
[4.1] http://www.active-endpoints.com/
[4.2] www.eclipse.org
[4.3] ActiveWebflow™ Enterprise and Standard Server User's Guide
[4.4] ActiveWebflow™ Designer User's Guide
[4.5] WfMC, *Workflow Reference Model*; available at: http://www.wfmc.org/standards/model.htm
[4.6] http://con-cern.org/
[4.7] http://projects.niekom.de/workflow/wiki/About/ SharkEnhydra
[4.8] http://shark.objectweb.org/doc/overview.html
[4.9] http://taverna.sourceforge.net/
[4.10] http://www.mygrid.org.uk
[4.11] http://tikiwiki.org/tiki-index.php?page= GalaxiaWorkflow
[4.12] http://bonita.objectweb.org/html/Documentation/ index.html
[4.13] http://www.inria.fr/recherche/equipes/ecoo.en.html
[4.14] IBM, BEA Systems, Microsoft, SAP AG, Siebel Systems: Business Process Execution Language for Web Services; Version 1.1; May 2003; available at: http://www-128.ibm.com/developerworks/library/specification/ws-bpel/
[4.15] OASIS: Web Services Business Process Execution Language; Committee Draft; Version 2.0; December 2005; available at: http://www.oasis-open.org/committees/download.php/16024/wsbpel-specification-draft-Dec-22-2005.htm
[4.16] Oracle Application Server Integration; Installation Guide; 10g Release 2 (10.1.2); August 2005; available at: http://www.oracle.com/technology/products/ias/bpel/index.html
[4.17] IBM, Microsoft: Web Services Description Language; Version 1.1; March 2001; available at: http://www.w3.org/TR/wsdl
[4.18] W3C: Web Services Addressing; W3C Member Submission; August 2004; available at: http://www.w3.org/Submission/ws-addressing/
[4.19] Oracle BPEL Process Manager; Developer's Guide; 10g Release 2 (10.1.2); October 2005; available at: http://www.oracle.com/technology/products/ias/bpel/index.html
[4.20] Apache Axis project; available at http://ws.apache.org/axis/
[4.21] Web Services Interoperability Organisation; Basic Profile Version 1.0; available at: http://www.ws-i.org/Profiles/BasicProfile-1.0-2004-04-16.html
[4.22] IBM, Microsoft: Web Services Inspection Language WS-Inspection; Version 1.0; 2001; available at: http://www-128.ibm.com/developerworks/library/specification/ws-wsilspec/
[4.23] Oracle Adapters for Files, FTP, Databases, and Enterprise Messaging; 10g Release 2 (10.1.2); November 2005; available at: http://www.oracle.com/technology/products/ias/bpel/index.html
[4.24] Apache: Web Services Invocation Framework; available at: http://ws.apache.org/wsif/
[4.25] http://www.webasyst.net/
[4.26] http://smarty.php.net
[4.27] http://www.wfmc.org/
[4.28] http://www.danet.com/
[4.29] http://schemas.xmlsoap.org/ws/2003/05/partner-link/
[5.1] http://info.lib.uh.edu/pr/v8/n3/smit8n3.html
[5.2] http://www.axia-consulting.co.uk/index.html
[5.3] http://www.library.cornell.edu/olinuris/ref/research/ webcrit.html
[5.4] http://www.cyberbee.com/content.pdf

[5.5] http://www.cyberbee.com/design.pdf
[5.6] http://www.standards.com.au/catalogue/script/ Search.asp
[5.7] http://jhss.wrdsb.on.ca/library/html/evaluate/ evalinfo.htm
[Aalst, 1998] W. M. P. van der Aalst, The application of Petri nets to workflow management. *The Journal of Circuits, Systems and Computers*, 8(1), p.21-66, 1998
[Aalst, 1999] W.M.P. van der Aalst, *Process-Oriented Architectures for Electronic Commerce and Interorganizational Workflow*. Information Systems, Vol. 24/8, 1999.
[Aalst et al., 2000] W.M.P. van der Aalst, A.H.M. ter Hofstede, B. Kiepuszewski, and A.P. Barros. Advanced Workflow Patterns. In O. Etzion and P. Scheuermann, editors, *7th International Conference on Cooperative Information Systems (CoopIS 2000)*, volume 1901 of *Lecture Notes in Computer Science*, pages 18-29. Springer-Verlag, Berlin, 2000.
[Aalst and Hee, 2002] W.M.P. van der Aalst, K.M. van Hee, Workflow Management: Models, Methods, and Systems. MIT press, Cambridge, MA, 2002.
[Aalst et al, 2002] W.M.P. van der Aalst, M. Dumas, A.H.M. ter Hofstede, and P. Wohed, Pattern-Based Analysis of BPML (and WSCI). QUT Technical report, FIT-TR-2002-05, Queensland University of Technology, Brisbane, 2002
[Aalst et al., 2003] W. M. P. van der Aalst, A. H. M. ter Hofstede, B. Kiepuszewski, and . P. Barros, Workflow patterns. D*istributed and Parallel Databases*, 14(1), .5-51, 2003
[Ader, 2004] Ader M., Workflow and Business Process Management Comparative Study, 2004
[Andrews, 2003] Andrews T., F. Curbera, H. Dholakia, Y. Goland, J. Klein, F. Leymann, K. Liu, D. Roller, D. Smith, S. Thatte, I. Trickovic, and S. Weerawarana, Business Process Execution Language for Web Services Version 1.1, 2003. Technical report, Accessed at http://xml.coverpages.org/ BPELv11-May052003Final.pdf
[Axis] Apache Axis project; available at http://ws.apache.org/axis/
[Belyk and Feist, 2002] Belyk, D. and D. Feist, (2002) Software Evaluation Criteria and Terminology. International Review of Research in Open and Distance Learning, ISSN: 1492-3831.
[BPEL-1.1] IBM, BEA Systems, Microsoft, SAP AG, Siebel Systems: Business Process Execution Language for Web Services; Version 1.1; May 2003; available at: http://www-128.ibm.com/developerworks/library /specification/ws-bpel/
[BPEL-2.0] OASIS: Web Services Business Process Execution Language; Committee Draft; Version 2.0; December 2005; available at: http://www.oasis-open.org/committees/download.php/16024/wsbpel-specification-draft-Dec-22-2005.htm
[BPEL-Ad] Oracle Adapters for Files, FTP, Databases, and Enterprise Messaging; 10g Release 2 (10.1.2); November 2005; available at: http://www.oracle.com/technology/products/ias/bpel/index.html
[BPEL-DG] Oracle BPEL Process Manager; Developer's Guide; 10g Release 2 (10.1.2); October 2005; available at: http://www.oracle.com/technology/products/ias/bpel/index.html
[BPEL-IG] Oracle Application Server Integration; Installation Guide; 10g Release 2 (10.1.2); August 2005; available at: http://www.oracle.com/technology/products/ias/bpel/index.html.
[BPEL-UniS] Hantschel, Ruf, Strotbek; Vergleich von BPEL Laufzeitum gebungen, Fachstudie 54 Universitat Stuttgart, January 2006; ftp://ftp.informatik.uni-stuttgart.de/pub/library/medoc.ustuttgart_fi/FACH-0054/FACH-0054.pdf (in German)

[BPMN] Business Process Modelling Notation (BPMN) Version 1.0; May 3, 2004; available at http://www.BPMI.ORG

[BPMI] Using BPMN to Model a BPEL Process; Stephen A. White; March 2005

[Cichocki, 1997] Cichocki A., A. Helal, M. Rusinkiewicz, and D.Woelk, *Workflow and Process Automation: Concepts and Technology*, Kluwer Academic Publishers, 1997, ISBN 0-7923-8099-1.

[Curbera, 2002] Curbera F., Y. Goland, J. Klein, F. Leymann, D. Roller, S. Thatte, S. Weerawarana, S. Thatte, Business Process Execution language for Web Services (Version 1.0), 2002. IBM

[Dayal, 2001] Dayal U., M. Hsu, R. Ladin, *Business Process Coordination - State of the Art, Trends, and Open Issues*, 2001. Proc. of the 27th VLDB Conference

[DBE06] DBE Execution Environment. European Project (2006). Vidal, M., Technical Coordinator.

[Ellis and Nutt, 1980] Ellis, C. A.; Nutt, G. J., Office Information Systems and Computer Science, 1980. In: ACM Computing Surveys, 12 (1980) 1, pp. 27-60.

[Fenton and Pfleeger , 1996] Fenton N. E. and S. L. Pfleeger, *Software Metrics: A Rigorous and Practical Approach,* 1996. Thomson Computer Press.

[Fil97] FileNet (1997). Visual WorkFlo Design Guide. FileNet Corporation, Costa Mesa, CA, USA.

[Fil99] FileNet (1999). Panagon Visual WorkFlo Architecture. FileNet Corporation, Costa Mesa, CA,USA.

[For98] Fort_e (1998). Fort_e Conductor Process Development Guide. Fort_e Software, Inc, Oakland, CA,USA.

[FRETRIS] R&D PROJECT INCO COPERNICUS Project 960183 – FRETRIS: Road Freight Transport Demand&Supply Information Systems – A Telematic Pilot Tool for Europe.

[Fuj99] Fujitsu (1999). i-Flow Developers Guide. Fujitsu Software Corporation, San Jose, CA, USA.

[Georgakopoulos *et al*., 1995] Georgakopoulos D., M. Hornick, A. Sheth. An Overview of Workflow Management: From Process Modelling to Workflow Automation Infrastructure, 1995, Kluwer Academic Publishers, Distributed and Parallel Databases, 3, 119-153

[Haller *et al*, 2005] Armin Haller, Eyal Oren, Simeon Petkov. Survey of Workflow Management Systems, May 2, 2005. Available at: http://www.m3pe.org/deliverables/syseval.pdf

[HP00] HP (2000). HP Changengine Process Design Guide. Hewlett-Packard Company, Palo Alto, CA,USA.

[IBM99] IBM (1999). IBM MQSeries Workow - Getting Started With Buildtime. IBM Deutschland Entwicklung GmbH, Boeblingen, Germany.

[Itpearls] Itpearls commerce Ag Homepage, Flyers, Papers and Documents from Itpearls about Process Modeler Version 2.0; available at: http://www.itp-commerce.com

[Jablonski *et al*, 1996]Jablonski S. and C. Bussler, *Workflow Management: Modelling Concepts, Architecture and Implementation,* 1996. International Thomson Computer Press, London.

[Kapoun, 1998] Kapoun, Jim (1988). "Teaching undergrads WEB evaluation: A guide for library instruction." *C&RL News* (July/August): 522-523.

[Kappel , 2000] Kappel G., S. Rausch-Schott, W. Retschitzegger, *A Framework for Workflow Management Systems Based on Objects, Rules and Roles,* 2000. ACM Computing Surveys Electronic Symposium on Object-Oriented Application Frameworks, M.E. Fayad (guest editor)

[Karen McLachlan, 2002a] Karen McLachlan, (2002a). WWW CYBERGUIDE RATINGS FOR CONTENT EVALUATION, East Knox High School Howard, Ohio, http://www.cyberbee.com/content.pdf

[Karen McLachlan, 2002b] Karen McLachlan, (2002b). WWW CYBERGUIDE RATINGS FOR WEB SITE DESIGN, East Knox High School Howard, Ohio , http://www.cyberbee.com/design.pdf

[Kiepuszewski, 2003] . Kiepuszewski, B. (2003) Expressiveness and Suitability of Languages for Control Flow Modelling in Workflows, PhD Thesis, Queensland University of Technology.

[Kitchenham, 1996] Kitchenham B. A., Evaluating Software Engineering Methods and Tool. Part 1: The Evaluation Context and the Evaluation Methods, 1996. *ACM SIGSOFT Software Engineering Notes*, 21, pp.11-15

[Leymann, 2000] Leymann F. and D. Roller, *Production workflow: concepts and techniques,* 2000. Prentice-Hall

[Leymann, 2001]Leymann F., Web Services Flow Language (WSFL 1.0), 2001, IBM

[Marshak, 1994] Marshak R. T., Perspectives on Workflow, In T. E. White, L. Fischer (ed.): *New Tools for New Times - The Workflow Paradigm*, 1994. Future Strategies, Inc., Alameda, CA, USA, pp 165-176.

[Mukherjee, *et al.*, 2004] Mukherjee S., *et al.*, Logic-based approaches to workflow modelling and verification. In J. Chomicki, R. van der Meyden, and G. Saake, (eds.) *Logics for Emerging Applications of Databases*, 2004 chap. 5. Springer-Verlag, Berlin, 167-202.

[Oestereich *et al.,* 2005] Bernd Oestereich, B., T. Weilkiens, F. Ferrandina, C. Bröcker iX Studie – Bessere Software – den Entwicklungsprozess im Griff UML2 und MDA Werkzeuge im Vergleich Heise Zeitschriften Verlag, November 2005 http://www.heise.de/kiosk/special/ixstudie/05/02/

[Peterson, 1981] Peterson, James Lyle. Petri Net Theory and the Modelling of Systems, Prentice Hall. ISBN 0136619835., 1981, p.288

[Reisig, 1992] Reisig, Wolfgang. A Primer in Petri Net Design, Springer-Verlag, ISBN 3-540-52044-9, 1992

[SAP97] SAP (1997). WF SAP Business Workow. SAP AG, Walldorf, Germany.

[Sheth, 1999] Sheth A. P., W. M. P. van der Aalst, I. B. Arpinar. Processes driving the networked economy, 1999. *IEEE Concurrency*, 7(3), 18-31.

[Smith, 1997] Smith, Alastair G., (1997) Testing the Surf: Criteria for Evaluating Internet Information Resources. *The Public-Access Computer Systems Review* 8, no. 3.

[Sta00] Sta_ware (2000). Sta_ware 2000 / GWD User Manual. Sta_ware plc, Berkshire, United Kingdom.

[Thatte, 2001] Thatte S., XLANG: Web Services for Business Process Design, 2001. Microsoft.

[Ver00] Verve (2000). Verve Component Workow Engine Concepts. Verve, Inc., San Francisco, CA,USA.

[Wegner, 1996] Wegner P., *Interoperability,* 1996. In: ACM Computing Surveys, Vol. 28, No. 1

[WfMC, Glossary, 1999] WfMC, Glossary, Terminology and Glossary, 3rd Edition. Document No WFMC-TC-1011. Workflow Management Coalition. Winchester, 1999.

[WfMC, 2002] WfMC, Workflow Process Definition Interface –XML Process Definition Language. Technical Report Document Number WFMC-TC-1025, Workflow Management Coalition, 2002, http://www.wfmc.org/standards/docs.htm

[Wohed, 2002] Wohed P., W.M.P. van der Aalst, M. Dumas, A.H.M. ter Hofstede, Pattern-Based Analysis of BPEL4WS. QUT Technical report, FIT-TR-2002-04, Queensland University of Technology, Brisbane, 2002.

[WSDL-1.1] IBM, Microsoft: Web Services Description Language; Version 1.1; March 2001; available at: http://www.w3.org/TR/wsdl

[WS-Addr] W3C: Web Services Addressing; W3C Member Submission; August 2004; available at: http://www.w3.org/Submission/ws-addressing/

[WS_I] Web Services Interoperability Organisation; Basic Profile Version 1.0; available at: http://www.ws-i.org/Profiles/BasicProfile-1.0-2004-04-16.html

[WS-Insp] IBM, Microsoft: Web Services Inspection Language WS-Inspection; Version 1.0; 2001; available at: http://www-128.ibm.com/developerworks/library/specification/ws-wsilspec/

[WS-Inv] Apache: Web Services Invocation Framework; available at: http://ws.apache.org/wsif/

[Yan *et al*., 2001] Y. Yan, Z. Maamar, W. Shen. Integration of Workflow and Agent Technology for Business Process Management. The Sixth International Conference on CSCW in Design. July 12-14, 2001, London, Ontario, Canada.

[zur Muehlen, 2004] M. zur Muehlen. Organizational management in workflow applications - issues and perspectives. *Information Technology and Management*, 5, 2004, 271-291

[zur Muehlen, 2004a] Michael zur Muehlen, Workflow-based Process Controlling. Foundation, Design, and Application of Workflow-driven Process Information Systems, 2004. Logos Verlag Berlin, 2004, ISBN 3-8325-0388-9.